The cyclic sliding operation in Garside groups

Volker Gebhardt[*] Juan González-Meneses[*,†]

September 6, 2008

Abstract

We present a new operation to be performed on elements in a Garside group, called cyclic sliding, which is introduced to replace the well known cycling and decycling operations. Cyclic sliding appears to be a more natural choice, simplifying the algorithms concerning conjugacy in Garside groups and having nicer theoretical properties. We show, in particular, that if a super summit element has conjugates which are *rigid* (that is, which have a certain particularly simple structure), then the optimal way of obtaining such a rigid conjugate through conjugation by positive elements is given by iterated cyclic sliding.

1 Introduction

Garside groups are a generalisation of Artin-Tits groups, hence of braid groups. Some theoretical and algorithmic problems related to conjugacy in these groups have been deeply studied, and several objects and tools have been defined and are well known to specialists. Among the best known tools are two special maps, called *cycling* and *decycling*, each of which sends a given element to some conjugate. These maps, introduced in [15], are the key ingredients for computing the so called *super summit sets* and *ultra summit sets*. The methods for computing these sets, their properties, and the properties of their elements constitute the main topic of research concerning problems in Garside groups related to conjugacy. [15, 8, 17, 19, 5, 6, 7, 22, 23, 24, 26]

In this paper we introduce a new operation, called *cyclic sliding*, and we propose to replace the usual cycling and decycling operations by this new one, as it is more natural from both the theoretical and computational points of view. Once cycling and decycling have been replaced by cyclic sliding, it is also natural to replace the ultra summit set $\mathrm{USS}(x)$ of an element x, whose definition is closely related to cycling, by its analogue for cyclic sliding. This set, called the *set of sliding circuits* and denoted $\mathrm{SC}(x)$, is the set of conjugates of x which are fixed points for some power of cyclic sliding. Obtaining an element in $\mathrm{SC}(x)$ starting from x merely requires applying iterated cyclic sliding until a repetition is encountered. We will see that $\mathrm{SC}(x)$ is a subset of $\mathrm{USS}(x)$ and that, like $\mathrm{USS}(x)$, it is a finite invariant set of the conjugacy class of x. The latter allows us to solve the conjugacy problem in Garside groups using $\mathrm{SC}(x)$ in place of $\mathrm{USS}(x)$.

The purpose of this paper is to emphasise the naturalness of the cyclic sliding operation, to stress how algorithms and proofs become in general simpler than the classical ones, and to show that the sets of sliding circuits and their elements naturally satisfy all the good properties that were already shown for ultra summit sets, in some cases having even better properties. For instance,

[*] Both authors partially supported by MTM2007-66929 and FEDER.

[†] This work was done partially while the second author was visiting the Institute for Mathematical Sciences, National University of Singapore in 2007. The visit was supported by the Institute.

for elements of canonical length 1, cycling and decycling are trivial operations, but cyclic sliding is not, and this allows us to extend some known results concerning conjugacy classes in a Garside group G to the case of canonical length 1. In particular, concerning rigid elements (see §4.2) we show:

Theorem 1.1. *Let G be a Garside group of finite type. If $x \in G$ is conjugate to a rigid element, then $\mathrm{SC}(x)$ is the set of rigid conjugates of x.*

The corresponding result for $\mathrm{SC}(x)$ replaced by $\mathrm{USS}(x)$ was known to hold if the elements in $\mathrm{USS}(x)$ have canonical length greater than 1 [5], but there are counterexamples if the elements in $\mathrm{USS}(x)$ have canonical length equal to 1. The use of cyclic sliding allows us to drop the condition on the canonical length and hence yields a conceptually simpler result by removing the need to consider special cases.

Still concerning rigid elements, we will prove the following result which shows, probably better than any other argument, why cyclic sliding is a natural choice:

Theorem 1.2. *If x is a super summit element that has rigid conjugates, then iterated cyclic sliding conjugates x to a rigid element and the obtained conjugating element is the minimal positive element doing so.*

One of the main advantages of considering the set $\mathrm{SC}(x)$ is that it yields a simpler algorithm to solve the conjugacy decision problem (to decide whether two elements are conjugate) and the conjugacy search problem (to find a conjugating element for two given conjugate elements) in Garside groups of finite type. The worst case complexity of this algorithm is not better than the previously known ones [19], but it is conceptually simpler and easier to implement. In this paper we give the idea of the algorithm; the details of the implementation and the study of complexity will be presented in [20].

In [5] the authors, together with Joan S. Birman, explained a project to solve the conjugacy problem in braid groups in polynomial time. This project, which was partially developed in [5, 6, 7], involved the concepts commonly used at that time, in particular ultra summit sets, cycling and decycling. We remark that the results in this paper do not modify the essential ideas in the above project: one just replaces ultra summit sets by sets of sliding circuits, and cycling and decycling by cyclic sliding. We believe this is a more natural and better way to look at the whole problem. The whole project and all the open problems can immediately be translated to this new setting and we believe that the latter will be a better point of view for solving the remaining open problems.

The structure of the paper is as follows. In Section 2 we give a basic introduction to the theory of Garside groups; specialists may skip this part, although the definition of local sliding in §2.2 should not be missed. In Section 3 we present the new concepts introduced in this paper: cyclic sliding in §3.1, the sets of sliding circuits in §3.2, the transport map in §3.3 and the sliding circuits graph in §3.4. Section 4 is devoted to theoretical applications of these new concepts: An algorithm to solve the conjugacy problem in Garside groups is explained in §4.1, applications to rigid elements – in particular the proofs of Theorems 1.1 and 1.2 – are given in §4.2, and finally we show in §4.3 that, in the particular case of the braid groups, the results which usually consider ultra summit sets to study reducible braids can also be translated to this new setting. Finally, Section 5 gives theoretical and computational examples comparing ultra summit sets to sets of sliding circuits in the case of braid groups.

Acknowledgements: Most of the ideas contained in this paper appeared in the framework of the collaboration of both authors and Joan S. Birman. We are grateful to her for so many discussions, for her advice and support. We also thank Patrick Dehornoy for useful conversations on Garside groups, specially for pointing out that Property (G4) of the definition of a Garside group needs to be checked only for the Garside element. We thank Pedro González Manchón for his comments on a previous draft of this paper, and for providing us with the example we treat in Section 5.

2 Background

2.1 Basic facts about Garside groups

Garside groups were defined by Dehornoy and Paris [12]. For a detailed introduction to these groups, see [13]; a shorter introduction, containing all the details needed for this paper can be found in [5] (§1.1 and the beginning of §1.2).

One of the possible definitions of a Garside group is the following. A group G is said to be a **Garside group** with **Garside structure** (G, P, Δ) if it admits a submonoid P satisfying $P \cap P^{-1} = \{1\}$, called the monoid of **positive elements**, and a special element $\Delta \in P$ called the **Garside element**, such that the following properties hold:

(G1) The partial order $\preccurlyeq$ defined on G by $a \preccurlyeq b \Leftrightarrow a^{-1}b \in P$ (which is invariant under left multiplication by definition) is a lattice order. That is, for every $a, b \in G$ there exist a unique least common multiple $a \vee b$ and a unique greatest common divisor $a \wedge b$ with respect to $\preccurlyeq$.

(G2) The set $[1, \Delta] = \{a \in G \mid 1 \preccurlyeq a \preccurlyeq \Delta\}$, called the set of **simple elements**, generates G.

(G3) Conjugation by Δ preserves P (so it preserves the lattice order $\preccurlyeq$). That is, $\Delta^{-1}P\Delta = P$.

(G4) For all $x \in P\backslash\{1\}$, one has:

$$||x|| = \sup\{k \mid \exists a_1, \ldots, a_k \in P\backslash\{1\} \text{ such that } x = a_1 \cdots a_k\} < \infty.$$

Definition 2.1. *A Garside structure* (G, P, Δ) *is said to be* **of finite type** *if the set of simple elements* $[1, \Delta]$ *is finite. A group* G *is said to be a* **Garside group of finite type** *if it admits a Garside structure of finite type.*

Throughout this paper, let G be a Garside group of finite type with a fixed Garside structure (G, P, Δ) of finite type.

Remarks:

1. By definition, $p \in P \Leftrightarrow 1 \preccurlyeq p$. This is why the elements of P are called positive. Given two positive elements $a \preccurlyeq b$, one usually says that a is a **prefix** of b. Hence the simple elements are the positive prefixes of Δ.

2. The number $||x||$ defined above for each $x \in P\backslash\{1\}$, defines a norm in P (setting $||1|| = 0$). Notice that the existence of this norm implies that every element in $P\backslash\{1\}$ can be written as a product of **atoms**, where an atom is an element $a \in P$ that cannot be decomposed in P, that is, $a = bc$ with $b, c \in P$ implies that either $b = 1$ or $c = 1$. In any decomposition of x as a product of $||x||$ factors in $P\backslash\{1\}$, all of them are atoms. Notice that the set of atoms generates G. Moreover, the set of atoms is finite if G is of finite type.

3. We learnt from Patrick Dehornoy that $||x|| < \infty$ for every $x \in P\backslash\{1\}$ if and only if $||\Delta|| < \infty$. Hence one does not need to check property (G4) for every positive element, but just for Δ.

The main examples of Garside groups of finite type are Artin-Tits groups of spherical type. In particular, braid groups are Garside groups. In the braid group B_n on n strands with the usual Garside structure that we call **Artin Garside structure** of B_n, one has the following:

- The atoms are the standard generators $\sigma_1, \ldots, \sigma_{n-1}$.

- The positive elements are the braids that can be written as a word which only contains positive powers of the atoms.
- The simple elements are the positive braids in which any two strands cross at most once. Here $|[1,\Delta]| = n!$, so this is a finite type Garside structure.
- The Garside element Δ (also called *half twist*) is the positive braid in which any two strands cross exactly once. That is, $\Delta = \sigma_1(\sigma_2\sigma_1)(\sigma_3\sigma_2\sigma_1)\cdots(\sigma_{n-1}\cdots\sigma_1)$.

It is important to note that in a Garside group, the monoid P induces not only a partial order $\preccurlyeq$ which is invariant under left multiplication, but also a partial order $\succcurlyeq$ which is invariant under right multiplication. The latter is defined by $a \succcurlyeq b \Leftrightarrow ab^{-1} \in P$. It is obvious from the definitions that $a \preccurlyeq b$ is equivalent to $a^{-1} \succcurlyeq b^{-1}$. It follows from the properties of G that $\succcurlyeq$ is also a lattice order, that P is the set of elements a such that $a \succcurlyeq 1$, and that the simple elements are the positive suffixes of Δ (where we say that a positive element b is a suffix of a if $a \succcurlyeq b$). We will denote by $x \wedge^{\Lsh} y$ (resp. $x \vee^{\Lsh} y$) the greatest common divisor (resp. least common multiple) of $x, y \in G$ with respect to $\succcurlyeq$.

The following notions are well known to specialists in Garside groups:

Definition 2.2. *Given a simple element s, the* **right complement** *of s is defined by $\partial(s) = s^{-1}\Delta$, and the* **left complement** *of s is $\partial^{-1}(s) = \Delta\, s^{-1}$.*

Notice that the map $\partial : [1,\Delta] \to [1,\Delta]$ is a bijection of the (finite) set $[1,\Delta]$. Notice also that $\partial^2(s) = \Delta^{-1}s\Delta$. We denote by τ the inner automorphism of G corresponding to conjugation by Δ. Hence $\partial^2(s) = \tau(s)$.

Definition 2.3. *Given two simple elements a and b, we say that the decomposition $a \cdot b$ is* **left weighted** *if $\partial(a) \wedge b = 1$ or, equivalently, if $ab \wedge \Delta = a$. We say that the decomposition $a \cdot b$ is* **right weighted** *if $a \wedge^{\Lsh} \partial^{-1}(b) = 1$ or, equivalently, if $ab \wedge^{\Lsh} \Delta = b$.*

Definition 2.4. *Given $x \in G$, we say that a decomposition $x = \Delta^p x_1 \cdots x_r$, where $p \in \mathbb{Z}$ and $r \geq 0$, is the* **left normal form** *of x if $x_i \in [1,\Delta]\backslash\{1,\Delta\}$ for $i = 1,\ldots,r$ and $x_i x_{i+1}$ is a left weighted decomposition for $i = 1,\ldots,r-1$. We say that a decomposition $x = y_1 \cdots y_r \Delta^p$ is the* **right normal form** *of x if $y_i \in [1,\Delta]\backslash\{1,\Delta\}$ for $i = 1,\ldots,r$ and $y_i y_{i+1}$ is a right weighted decomposition for $i = 1,\ldots,r-1$.*

It is well known that left and right normal forms of elements in G exist and are unique. Moreover, the numbers p and r do not depend on the normal form (left or right) that we are considering.

Definition 2.5. *Given $x \in G$, whose left normal form is $\Delta^p x_1 \cdots x_r$ and whose right normal form is $y_1 \cdots y_r \Delta^p$, we define the* **infimum**, **canonical length** *and* **supremum** *of x, respectively, by $\inf(x) = p$, $\ell(x) = r$ and $\sup(x) = p + r$.*

It is shown in [15] that $\inf(x)$ and $\sup(x)$ are precisely the maximal and minimal integers, respectively, such that $\Delta^{\inf(x)} \preccurlyeq x \preccurlyeq \Delta^{\sup(x)}$ (or, equivalently, $\Delta^{\sup(x)} \succcurlyeq x \succcurlyeq \Delta^{\inf(x)}$).

2.2 Left normal forms and local sliding

The definition of cyclic sliding in G will appear to be a natural notion once we notice how normal forms in G are computed. This is what we recall in this subsection.

Recall that given two positive elements $a, c \in P$, one has $a \preccurlyeq c$ if and only if c can be written as $c = ab$, where $b \in P$. This is why a is said to be a **prefix** of c in this case. This allows to describe the left weightedness of a decomposition (and hence left normal forms) in a particularly simple way.

As we saw in Definition 2.3, given two simple elements a and b, the decomposition ab is said to be left weighted if $\partial(a) \wedge b = 1$, that is, if $\partial(a)$ and b have no prefixes in common (except the trivial one). Since $\partial(a)$ is the simple element such that $a\,\partial(a) = \Delta$, the prefixes of $\partial(a)$ are precisely the simple elements s such that as is a prefix of Δ, or in other words, such that as is simple. Therefore, the decomposition ab is left weighted if and only if the only prefix s of b such that as is simple is the trivial one.

Using this description of left weightedness, it is easy to give a procedure to find the left weighted factorisation of the product of two simple elements a and b as follows. If the decomposition ab is not left weighted, this means that there is a nontrivial prefix $s \preccurlyeq b$ such that as is simple (i.e. $s \preccurlyeq \partial(a)$). Since $\preccurlyeq$ is a lattice order, there is a maximal element satisfying the above property, namely $s = \partial(a) \wedge b$. Therefore, the only thing to do in order to transform the decomposition ab into a left weighted one, is to *slide* the prefix $s = \partial(a) \wedge b$ from the second factor to the first one. That is, write $b = st$ and then consider the decomposition $ab = (as)t$, with (as) as the first factor and t as the second one. The decomposition $(as)t$ is left weighted by the maximality of s (alternatively, multiplying the equation $\partial(a) \wedge b = s$ on the left by s^{-1} one obtains $\partial(as) \wedge t = 1$).

The action of transforming the decomposition $ab = a(st)$ into the left weighted decomposition $(as)t$, by *sliding* the simple element s from the second factor to the first factor, will be called a **local sliding** applied to the decomposition ab (see Figure 1).

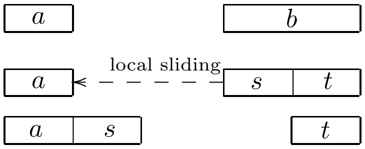

Figure 1: Local sliding of ab, where a and b are simple. The slid element is $s = \partial(a) \wedge b$. The decomposition ab is not necessarily left weighted, but $(as)t$ is.

Using local slidings one can compute the left normal form for every element of a Garside group G. This normal form follows ideas from Garside [18] and was defined in [14, 1, 15, 16] in the case of braid groups. The same notion extends to every Garside group and is the basis of Definition 2.4. To see how we can compute a left normal form using local slidings, let $x \in G$ be written as a product of simple elements and their inverses, that is, $x = s_1^{e_1} \cdots s_m^{e_m}$, where every s_i is simple and $e_i = \pm 1$. Replace each s_i^{-1} by $\partial(s_i)\Delta^{-1}$ and then collect all the appearances of $\Delta^{\pm 1}$ on the left, applying τ or τ^{-1}. In this way one obtains $x = \Delta^q t_1 \cdots t_k$, where $q \in \mathbb{Z}$ and every t_i is simple. Then one just needs to apply a local sliding to any pair of consecutive factors and keep doing this until all consecutive factors are left weighted. In this way, all appearances of Δ will be collected on the left (this increases the power q), and all appearances of the trivial element will be collected on the right (and one can erase them). This yields $x = \Delta^p x_1 \cdots x_r$ written in left normal form.

During the process of computing local slidings to obtain a left normal form, it is convenient to know the following result. It says that if a product of k simple elements is already in left normal form, and we multiply it (either from the left or from the right) by a simple element, then one can obtain the left normal form of the product by applying only k local slidings.

Proposition 2.6 (see, for example, [11, Props. 3.1 and 3.3] or [16])**.** *Let $s_1, \ldots, s_k$ and s_0', s_{k+1}' be simple elements such that the product $s_1 \cdots s_k$ is in left normal form as written.*

1. *Consider the product $s_0' s_1 \cdots s_k$. For $i = 1, \ldots, k$ apply a local sliding to the pair $s_{i-1}' s_i$, that is, let $t_i = \partial(s_{i-1}') \wedge s_i$ and define $s_{i-1}'' = s_{i-1}' t_i$ and $s_i' = t_i^{-1} s_i$. Finally define $s_k'' = s_k'$. Then $s_0'' \cdots s_k''$ is the left normal form of $s_0' s_1 \cdots s_k$ (where possibly $s_0'' = \Delta$ or $s_k'' = 1$).*

2. *Consider the product $s_1 \cdots s_k s'_{k+1}$. For $i = k, \ldots, 1$ apply a local sliding to the pair $s_i s'_{i+1}$, that is, let $t_i = \partial(s_i) \wedge s'_{i+1}$ and define $s'_i = s_i t_i$ and $s''_{i+1} = t_i^{-1} s'_{i+1}$. Finally define $s''_1 = s'_1$. Then $s''_1 \cdots s''_{k+1}$ is the left normal form of $s_1 \cdots s_k s'_{k+1}$ (where possibly $s''_1 = \Delta$ or $s''_{k+1} = 1$).*

It is known that if $x = \Delta^p x_1 \cdots x_r$ is in left normal form, then p is maximal and r is minimal among all possible decompositions of x as a power of Δ times a product of simple elements. We recall from Definition 2.5 that the number p is called the **infimum** of x, denoted $\inf(x)$, the number r of non-Δ factors is called the **canonical length** of x, written $\ell(x)$, and the sum $p+r$ of these two numbers is called the **supremum** of x, denoted $\sup(x)$. Notice that $1 \preccurlyeq x_1 \cdots x_r \preccurlyeq \Delta^r$. Multiplying on the left by Δ^p, one has $\Delta^{\inf(x)} \preccurlyeq x \preccurlyeq \Delta^{\sup(x)}$, where $\inf(x)$ and $\sup(x)$ are, respectively, the maximal and minimal numbers satisfying the above inequality [15]. The canonical length $\ell(x) = \sup(x) - \inf(x)$ is a length function $\ell : G \to \mathbb{N}$ that measures, in some sense, the complexity of the elements in G.

Let x^G denote the conjugacy class of x in G and define $\inf_s(x) = \max\{\inf(y) \mid y \in x^G\}$, $\sup_s(x) = \min\{\sup(y) \mid y \in x^G\}$ and $\ell_s(x) = \min\{\ell(y) \mid y \in x^G\}$. It is shown in [15] that the maximum of the infimum and the minimum of the supremum on x^G can be achieved simultaneously, whence $\ell_s(x) = \sup_s(x) - \inf_s(x)$. The so-called **super summit set** $\mathrm{SSS}(x)$ of x, defined as the set of conjugates of x with maximal infimum and minimal supremum (and hence with minimal canonical length)

$$\mathrm{SSS}(x) = \{y \in x^G \mid \inf(y) = \inf_s(x) \text{ and } \sup(y) = \sup_s(x)\} = \{y \in x^G \mid \ell(y) = \ell_s(x)\}$$

is hence non-empty. As G is of finite type, the set $\mathrm{SSS}(x)$ is finite.

To end this section, we will compare the left normal forms of x and x^{-1} in a similar way as it is done in [15]. Notice that, by definition, a product ab is left weighted if and only if the product $\partial^{-1}(b)\,\partial(a)$ is left weighted. Since $\partial^2 \equiv \tau$, and τ preserves the order $\preccurlyeq$, it follows that ab is left weighted if and only if $\partial^{2k-1}(b)\,\partial^{2k+1}(a)$ is left weighted for every $k \in \mathbb{Z}$. From this it is obvious that if $x = \Delta^p x_1 \cdots x_r$ is in left normal form, the left normal form of x^{-1} is given by

$$x^{-1} = \Delta^{-(p+r)}\,\partial^{-2(p+r)+1}(x_r)\,\partial^{-2(p+r-1)+1}(x_{r-1}) \cdots \partial^{-2(p+1)+1}(x_1)\,.$$

In particular, $\inf(x^{-1}) = -\sup(x)$, $\sup(x^{-1}) = -\inf(x)$ and $\ell(x^{-1}) = \ell(x)$. Note that this implies in particular that $\inf_s(x^{-1}) = -\sup_s(x)$, $\sup_s(x^{-1}) = -\inf_s(x)$, $\ell_s(x^{-1}) = \ell_s(x)$ and $\mathrm{SSS}(x^{-1}) = \{y^{-1} \mid y \in \mathrm{SSS}(x)\}$.

3 New concepts

3.1 Cyclic sliding

The usual algorithms to solve the conjugacy problem in Garside groups [18, 15, 8, 17, 19] share a common basic strategy. Given an element $x \in G$, the idea is to compute a finite subset of the conjugacy class x^G of x, which consists of those conjugates satisfying some suitable conditions, and which only depends on x^G, not on x itself. The particular subset used and the way in which it is computed differ for each one of the above algorithms. In this paper we define a new subset of x^G, which is different from (and smaller than) the above ones.

The main idea is the following. Given a product $x = \Delta^p x_1 \cdots x_r$ of simple elements, one may wonder how it can be *simplified*, for example in terms of reducing the number of factors. Using local slidings, one may be able to reduce the number of factors as discussed above, resulting in the left normal form of x. Recall that the left normal form of x is the simplest possible one in terms of the required number of factors, so one cannot expect to simplify it any more by local slidings, since each pair of consecutive factors is left weighted. However, we can look at x up to conjugacy; this is like looking at its factors written around a circle, so the factors x_r and x_1 can then be thought of as being consecutive, up to conjugacy (actually, there is Δ^p between them, but one can move it out of the way using τ).

One can then make x_r and x_1 interact by a suitable conjugation and try to simplify the obtained element using local slidings. This was the idea in [15], where ElRifai and Morton defined **cycling** and **decycling** in the following way.

Definition 3.1. [15] *Given $x = \Delta^p x_1 \cdots x_r$ written in left normal form, where $r > 0$, one defines the following conjugates of x: The* **cycling** *of x,*

$$\mathbf{c}(x) = x^{\tau^{-p}(x_1)} = \Delta^p x_2 \cdots x_r \, \tau^{-p}(x_1),$$

and the **decycling** *of x,*

$$\mathbf{d}(x) = x^{x_r^{-1}} = x_r \Delta^p x_1 \cdots x_{r-1}.$$

Roughly speaking, cycling moves the first factor to the back, whereas decycling moves the final factor to the front. Then one can use local slidings again, and the element we started with will possibly be simplified. It is obvious from the definition that neither cycling nor decycling can increase the canonical length of an element. One may hope that using iterated cycling and decycling one can find an element of minimal canonical length in the conjugacy class of x. This is actually the case, as shown in [15], but one needs to apply both kinds of conjugation, and one must use some results in [9] to know where to stop using one of them and start using the other one. Now we will present a single kind of conjugation, which will simplify the original element as much as possible in a very easy way.

We will assume for a moment that $x = \Delta^p x_1 \cdots x_r$ is in left normal form and $r > 1$. In order to make the last and the first factors of x interact, we can write $x = \tau^{-p}(x_1)\Delta^p x_2 \cdots x_r$. Considering this element up to conjugacy, the factors x_r and $\tau^{-p}(x_1)$ can be thought of as being consecutive and we may try to decompose the product $x_r \, \tau^{-p}(x_1)$ in a left weighted manner. That is, we want to apply a local sliding to $x_r \, \tau^{-p}(x_1)$. As we saw above, this is done by considering the simple element $s = \partial(x_r) \wedge \tau^{-p}(x_1)$. Then, if we write $\tau^{-p}(x_1) = s\,t$ one has $x_r \, \tau^{-p}(x_1) = (x_r s)\, t$, where the latter decomposition is left weighted. Therefore, s is the prefix of $\tau^{-p}(x_1)$ that should be *slid* to be multiplied on the right to x_r. If we recall that $x = \tau^{-p}(x_1)\Delta^p x_2 \cdots x_r$, this means that, in order to simplify the pair formed by the last and the first factors of x, one should remove the prefix s from $\tau^{-p}(x_1)$ and to multiply it to x_r from the right. In other words, one should conjugate x by s. This is what we will call a *cyclic sliding*. This can equally be defined for elements of canonical length 1, and it can be thought of as a trivial conjugation for elements of canonical length 0.

In order to give a more elegant definition, we recall from [5] that the **initial factor** $\iota(x)$ of an element $x \in G$ is defined as $\iota(x) = x\Delta^{-\inf(x)} \wedge \Delta$ and that the **final factor** $\varphi(x)$ of x is defined as $\varphi(x) = (\Delta^{\sup(x)-1} \wedge x)^{-1}\, x$. If $\ell(x) = r > 0$ and $x = \Delta^p x_1 \cdots x_r$ is in left normal form, the above definitions mean $\iota(x) = \tau^{-p}(x_1)$ and $\varphi(x) = x_r$, whereas for $\ell(x) = 0$ one has $\iota(x) = 1$ and $\varphi(x) = \Delta$. We also recall that, due to the relation between the left normal forms of x and x^{-1}, one has $\partial(\varphi(x)) = \iota(x^{-1})$ for every $x \in G$. Hence, the conjugating element s defined above is precisely $s = \iota(x) \wedge \partial(\varphi(x)) = \iota(x) \wedge \iota(x^{-1})$. This is a very particular prefix of x (and also of x^{-1}), so we give it a name.

Definition 3.2. *Given $x \in G$, we define the* **preferred prefix** *of x to be $\mathfrak{p}(x) = \iota(x) \wedge \iota(x^{-1})$. Or, equivalently, $\mathfrak{p}(x) = \left(x\Delta^{-\inf(x)}\right) \wedge \left(x^{-1}\Delta^{\sup(x)}\right) \wedge \Delta$.*

We can finally define our desired special conjugation:

Definition 3.3. *Given $x \in G$, we define the* **cyclic sliding** $\mathfrak{s}(x)$ *of x as the conjugate of x by its preferred prefix, that is,*

$$\mathfrak{s}(x) = x^{\mathfrak{p}(x)} = \mathfrak{p}(x)^{-1} x \, \mathfrak{p}(x).$$

Notice that $\mathfrak{p}(x)$ is precisely the element s defined above. We remark that the definition of $\mathfrak{p}(x)$ is invariant under taking inverses, that is, $\mathfrak{p}(x) = \mathfrak{p}(x^{-1})$, whence $\mathfrak{s}(x)^{-1} = \mathfrak{s}(x^{-1})$. It also follows immediately from the definitions that $\mathfrak{p}(\tau(x)) = \tau(\mathfrak{p}(x))$, whence $\mathfrak{s}(\tau(x)) = \tau(\mathfrak{s}(x))$.

Recall that we defined cyclic sliding in order to try to simplify the complexity of a given element. The following results show that cyclic sliding indeed results in a simplification.

Lemma 3.4. *For every $x \in G$, one has the inequalities*

1. $\inf(\mathfrak{s}(x)) \ \geq \ \inf(x)$
2. $\sup(\mathfrak{s}(x)) \ \leq \ \sup(x)$
3. $\ell(\mathfrak{s}(x)) \ \leq \ \ell(x)$

Proof. If $\ell(x) = 0$ then $\mathfrak{s}(x) = x$ and the result is clear. Otherwise, let $\Delta^p x_1 \cdots x_r$ be the left normal form of x. Since $\mathfrak{p}(x)$ is a prefix of $\iota(x) = \tau^{-p}(x_1)$, one can decompose $\tau^{-p}(x_1) = \mathfrak{p}(x)\, t$ for some simple element t. Since $\mathfrak{p}(x)$ is also a prefix of $\iota(x^{-1}) = \partial(x_r)$, the element $x_r\mathfrak{p}(x)$ is simple. Therefore, in the case $r > 1$ one has $\mathfrak{s}(x) = (\mathfrak{p}(x)\, t\, \Delta^p x_2 \cdots x_r)^{\mathfrak{p}(x)} = \Delta^p\, \tau^p(t)\, x_2 \cdots x_{r-1}\, (x_r\mathfrak{p}(x))$, where each factor in the latter decomposition is simple. It follows that $\Delta^p \preccurlyeq \mathfrak{s}(x) \preccurlyeq \Delta^{p+r}$, which implies the result. If $r = 1$ then $\mathfrak{s}(x) = \Delta^p(\tau^p(t)\mathfrak{p}(x))$, where the non-$\Delta$ factor is simple as it is a suffix of $x_1\mathfrak{p}(x)$ (recall that in this case $x_1 = x_r$). Hence the result also holds in this case. □

Corollary 3.5. *For every $x \in G$, iterated application of cyclic sliding eventually reaches a period, that is, there are integers $N \geq 0$ and $M > 0$ such that $\mathfrak{s}^{M+N}(x) = \mathfrak{s}^N(x)$.*

Proof. By Lemma 3.4 there is an integer K such that for all $k \geq K$ we have $\inf(\mathfrak{s}^k(x)) = \inf(\mathfrak{s}^K(x))$ and $\sup(\mathfrak{s}^k(x)) = \sup(\mathfrak{s}^K(x))$: Indeed, as $\inf(y) \leq \sup(y)$ for every element $y \in G$, $\inf(\mathfrak{s}^k(x))$ can only increase and $\sup(\mathfrak{s}^k(x))$ can only decrease a finite number of times. This implies that for all $k \geq K$ we also have $\ell(\mathfrak{s}^k(x)) = \ell(\mathfrak{s}^K(x))$. As G is of finite type, the set of elements with given infimum and canonical length is finite, which implies the claim. □

As in the case of cycling and decycling, one may hope that once a period under iterated cyclic sliding is reached, the canonical length of the involved element is minimal in the conjugacy class, that is, that iterated cyclic sliding decreases the canonical length to its minimal possible value. This is actually true. It is a direct consequence of the following results, in which we compare cyclic sliding with cycling and decycling.

Lemma 3.6. *For any $x \in G$ one has the following:*

1. *$\varphi(x)\iota(x) \preccurlyeq \Delta$ if and only if $\mathfrak{p}(x) = \iota(x)$. In this case, $\mathfrak{s}(x) = \mathbf{c}(x)$.*
2. *$\Delta \preccurlyeq \varphi(x)\iota(x)$ if and only if $\mathfrak{p}(x) = \iota(x^{-1}) = \varphi(x)^{-1}\Delta$. In this case, $\mathfrak{s}(x) = \tau(\mathbf{d}(x))$.*

Proof. Recall that $\mathbf{c}(x) = x^{\iota(x)}$ and $\mathbf{d}(x) = x^{\varphi(x)^{-1}}$. One has $\varphi(x)\iota(x) \preccurlyeq \Delta$ if and only if $\iota(x) \preccurlyeq \partial(\varphi(x)) = \iota(x^{-1})$, which in turn is equivalent to $\iota(x) = \iota(x) \wedge \iota(x^{-1}) = \mathfrak{p}(x)$. Hence claim 1 holds. Similarly, $\Delta \preccurlyeq \varphi(x)\iota(x)$ if and only if $\partial(\varphi(x)) \preccurlyeq \iota(x)$, which in turn is equivalent to $\varphi(x)^{-1}\Delta = \partial(\varphi(x)) = \iota(x) \wedge \partial(\varphi(x)) = \iota(x) \wedge \iota(x^{-1}) = \mathfrak{p}(x)$. Hence claim 2 holds. □

Lemma 3.7. *For any $x \in G$ with canonical length $\ell(x) > 1$ one has the following:*

1. *If $\varphi(x)\iota(x) \not\preccurlyeq \Delta$, then $\mathfrak{s}(x) = \mathbf{d}(\mathbf{c}(x))$.*
2. *If $\Delta \not\preccurlyeq \varphi(x)\iota(x)$, then $\mathfrak{s}(x) = \mathbf{c}(\mathbf{d}(x))$.*

Proof. If $\varphi(x)\iota(x) \not\preccurlyeq \Delta$, Lemma 3.6 yields $\mathfrak{p}(x) \prec \iota(x)$, that is, $\iota(x) = \mathfrak{p}(x)\, s$ for some non-trivial simple element s. If $x = \Delta^p x_1 \cdots x_r$ in left normal form, where $r > 1$, $\iota(x) = \tau^{-p}(x_1)$ and $\varphi(x) = x_r$, then $\mathbf{c}(x) = \Delta^p x_2 \cdots x_{r-1} \varphi(x)\iota(x) = \Delta^p x_2 \cdots x_{r-1}(\varphi(x)\mathfrak{p}(x))s$, where $(\varphi(x)\mathfrak{p}(x))s$ is left weighted. As $\Delta^p x_2 \cdots x_{r-1}\varphi(x)$ is in left normal form, this in particular implies $\varphi(\mathbf{c}(x)) = s$ (see Proposition 2.6). Hence, $\mathbf{d}(\mathbf{c}(x)) = (\mathbf{c}(x))^{s^{-1}} = x^{\iota(x)s^{-1}} = x^{\mathfrak{p}(x)} = \mathfrak{s}(x)$.

If $\Delta \not\preccurlyeq \varphi(x)\iota(x)$, then $\iota(x^{-1}) = \partial(\varphi(x)) \not\preccurlyeq \iota(x)$, whence $\varphi(x^{-1})\iota(x^{-1}) \not\preccurlyeq \varphi(x^{-1})\iota(x) = \Delta$. As one has $\mathbf{c}(y^{-1}) = \tau(\mathbf{d}(y))^{-1}$ as well as $\mathbf{c}(\tau(y)) = \tau(\mathbf{c}(y))$ and $\mathbf{d}(\tau(y)) = \tau(\mathbf{d}(y))$ for every $y \in G$, using the first claim for x^{-1} yields $\mathfrak{s}(x)^{-1} = \mathfrak{s}(x^{-1}) = \mathbf{d}(\mathbf{c}(x^{-1})) = \mathbf{d}(\tau(\mathbf{d}(x))^{-1}) = \tau^{-1}(\mathbf{c}(\tau(\mathbf{d}(x))))^{-1} = \mathbf{c}(\mathbf{d}(x))^{-1}$, that is, $\mathfrak{s}(x) = \mathbf{c}(\mathbf{d}(x))$. □

Lemma 3.8. *Let $x \in G$ with canonical length $\ell(x) > 1$.*

1. *If $\Delta \preccurlyeq \varphi(x)\iota(x) \preccurlyeq \Delta$, then $\mathfrak{s}(x) = \tau(\mathbf{d}(x)) = \mathbf{c}(x)$ and $\ell(\mathfrak{s}(x)) < \ell(x)$.*
2. *If $\Delta \not\preccurlyeq \varphi(x)\iota(x) \preccurlyeq \Delta$, then $\mathfrak{s}(x) = \mathbf{c}(\mathbf{d}(x)) = \mathbf{c}(x)$ and $\ell(\mathfrak{s}(x)) < \ell(x)$.*
3. *If $\Delta \preccurlyeq \varphi(x)\iota(x) \not\preccurlyeq \Delta$, then $\mathfrak{s}(x) = \tau(\mathbf{d}(x)) = \mathbf{d}(\mathbf{c}(x))$ and $\ell(\mathfrak{s}(x)) < \ell(x)$.*
4. *If $\Delta \not\preccurlyeq \varphi(x)\iota(x) \not\preccurlyeq \Delta$, then $\mathfrak{s}(x) = \mathbf{c}(\mathbf{d}(x)) = \mathbf{d}(\mathbf{c}(x))$.*

Moreover, if $\ell(\mathbf{c}(\mathbf{d}(x))) = \ell(x)$ or $\ell(\mathbf{d}(\mathbf{c}(x))) = \ell(x)$ then case 4 applies, which in particular implies $\mathbf{d}(\mathbf{c}(x)) = \mathbf{c}(\mathbf{d}(x))$.

Proof. The claimed equalities for $\mathfrak{s}(x)$ follow from Lemma 3.6 and Lemma 3.7. In cases 1 and 2 one has $\sup(\mathfrak{s}(x)) = \sup(\mathbf{c}(x)) \leq \sup(x) - 1$, whereas in cases 1 and 3 one has $\inf(\mathfrak{s}(x)) = \inf(\mathbf{d}(x)) \geq \inf(x)+1$. The last statement then follows since $\ell(\mathbf{d}(\mathbf{c}(x))) \leq \ell(\mathbf{c}(x))$ and $\ell(\mathbf{c}(\mathbf{d}(x))) \leq \ell(\mathbf{d}(x))$. □

Corollary 3.9. *For every $x \in G$, if $\ell(x)$ is not minimal in the conjugacy class of x, then there exists a positive integer $m < ||\Delta||$ such that $\ell(\mathfrak{s}^m(x)) < \ell(x)$.*

Proof. It is shown in [9, Theorem 1] that $\inf(\mathbf{c}^{||\Delta||-1}(x)) = \inf(x)$ implies $\inf(x) = \inf_s(x)$ and that $\sup(\mathbf{d}^{||\Delta||-1}(x)) = \sup(x)$ implies $\sup(x) = \sup_s(x)$. As cycling and decycling are trivial modulo τ for elements of canonical length 0 or 1, this in particular implies that if an element has canonical length 0 or 1, then this canonical length is already minimal in its conjugacy class.

Let then $m = ||\Delta|| - 1$ and assume that $\ell(\mathfrak{s}^i(x)) = \ell(x) = r > 1$ for $i = 1, \ldots, m$. In particular, for each $i = 0, \ldots, m-1$, one has $\ell(\mathfrak{s}(\mathfrak{s}^i(x))) = \ell(\mathfrak{s}^i(x))$, that is, the element $\mathfrak{s}^i(x)$ falls in the case 4 of Lemma 3.8, which implies $\mathfrak{s}(\mathfrak{s}^i(x)) = \mathbf{c}(\mathbf{d}(\mathfrak{s}^i(x))) = \mathbf{d}(\mathbf{c}(\mathfrak{s}^i(x)))$. Hence, $\mathfrak{s}^m(x) = \mathbf{c}\circ\mathbf{d}\circ\ldots\circ\mathbf{c}\circ\mathbf{d}(x)$, where the last expression involves m cyclings and m decyclings. Moreover, again by Lemma 3.8, each occurrence of $\mathbf{d} \circ \mathbf{c}$ can be replaced by $\mathbf{c} \circ \mathbf{d}$, or vice versa, since all intermediate elements have canonical length r. Repeating this argument, one obtains $\mathfrak{s}^m(x) = \mathbf{c}^m(\mathbf{d}^m(x)) = \mathbf{d}^m(\mathbf{c}^m(x))$.

As neither cycling nor decycling can increase the canonical length, the last equalities imply $\ell(\mathbf{d}^m(x)) = r = \ell(x)$ and $\ell(\mathbf{c}^m(x)) = r = \ell(x)$, which by [9] yield $\sup(x) = \sup_s(x)$ and $\inf(x) = \inf_s(x)$, that is, x has minimal canonical length in its conjugacy class. □

Corollary 3.10. *Let $x \in G$ with $\ell(x) = r$. There exists an integer $M \leq (r-1)(||\Delta|| - 1)$, such that $\mathfrak{s}^m(x) \in \mathrm{SSS}(x)$ for all $m \geq M$.*

Proof. The sequence $(\ell(\mathfrak{s}^m))_{m \in \mathbb{N}}$ is bounded below and monotonically decreasing by Lemma 3.6, so it stabilises; say at $m = M$. By Corollary 3.9, this means $\mathfrak{s}^m(x) \in \mathrm{SSS}(x)$ for all $m \geq M$. Since elements of canonical length 1 have minimal canonical length in their conjugacy class, the sequence can decrease at most $r - 1$ times, with at most $||\Delta|| - 1$ applications of cyclic sliding between any two decreases, again by Corollary 3.9. Hence, $M \leq (r-1)(||\Delta|| - 1)$ as claimed. □

Notice that the above results yield a very easy algorithm to produce a super summit conjugate of an element $x \in G$: Apply iterated cyclic sliding to x; if the canonical length of the resulting elements does not decrease within $||\Delta|| - 1$ consecutive applications, the super summit set has been reached.

3.2 The set of sliding circuits

After the results from the previous section it is clear that cyclic sliding, in some sense, reduces the complexity of a given element x. However, it is well known that the set of conjugates of x of minimal canonical length, that is the super summit set [15], can be a huge set. A much smaller set was defined in [19] using cycling, and in [25] using cycling and decycling. We will parallel those constructions here using cyclic sliding. More precisely, the idea is to continue applying iterated cyclic sliding until one obtains a repeated element, say y, and consider the resulting element y as one of those having the *best possible properties*, or at least the best possible properties that can be achieved using cyclic sliding. Notice that all elements in the orbit of y under cyclic sliding will also satisfy the same condition. We will say that such elements belong to a *sliding circuit* (the terminology comes from graph theory, since we will use graphs to study this situation, as we shall see). But in the conjugacy class of x there can be other sliding circuits, apart from the one containing y. We must then consider all of them, in order to obtain an invariant subset of the conjugacy class of x. This is done as follows.

Definition 3.11. *Given $y \in G$, we say that y belongs to a* **sliding circuit** *if $\mathfrak{s}^m(y) = y$ for some $m \geq 1$. Given $x \in G$, we define the* **set of sliding circuits of** *x, denoted by $\mathrm{SC}(x)$, as the set of all conjugates of x which belong to a sliding circuit.*

It is clear by definition that $\mathrm{SC}(x)$ does not depend on x but only on its conjugacy class. Hence, two elements $x, y \in G$ are conjugate if and only if $\mathrm{SC}(x) = \mathrm{SC}(y)$ or, equivalently, $\mathrm{SC}(x) \cap \mathrm{SC}(y) \neq \emptyset$. In particular, the computation of $\mathrm{SC}(x)$ and of one element of $\mathrm{SC}(y)$ will solve the conjugacy decision problem in G.

The strategy of defining a finite invariant subset of the conjugacy class has been used several times in the literature. The main well known examples, together with the set $\mathrm{SC}(x)$ we just defined, are the following:

Definition 3.12. *Given $x \in G$, we define the following subsets of the conjugacy class x^G of x:*

- The **summit set** of x [18],
$$\mathrm{SS}(x) \quad = \quad \{y \in x^G \mid \inf(y) \text{ is maximal in } x^G\}.$$
- The **super summit set** of x [15],
$$\begin{aligned}\mathrm{SSS}(x) \quad &= \quad \{y \in x^G \mid \ell(y) \text{ is minimal in } x^G\} \\ &= \quad \{y \in x^G \mid \inf(y) \text{ is maximal and } \sup(y) \text{ is minimal in } x^G\}.\end{aligned}$$
- The **ultra summit set** of x [19],
$$\mathrm{USS}(x) \quad = \quad \{y \in \mathrm{SSS}(x) \mid \mathbf{c}^m(y) = y \text{ for some } m \geq 1\}.$$
- The **reduced super summit set** of x [25],
$$\mathrm{RSSS}(x) \quad = \quad \{y \in x^G \mid \mathbf{c}^m(y) = y \text{ and } \mathbf{d}^n(y) = y \text{ for some } m, n \geq 1\}.$$
- The **set of sliding circuits** of x,
$$\mathrm{SC}(x) \quad = \quad \{y \in x^G \mid \mathfrak{s}^m(y) = y \text{ for some } m \geq 1\}.$$

The relation between all these sets is given by the following result.

Proposition 3.13. *Given $x \in G$, one has:*

$$\mathrm{SC}(x) \subseteq \mathrm{RSSS}(x) \subseteq \mathrm{USS}(x) \subseteq \mathrm{SSS}(x) \subseteq \mathrm{SS}(x).$$

Moreover, if $\ell_s(x) > 1$ then

$$\mathrm{SC}(x) = \mathrm{RSSS}(x),$$

and if $\ell_s(x) = 1$ then

$$\mathrm{SC}(x) \subseteq \mathrm{RSSS}(x) = \mathrm{USS}(x) = \mathrm{SSS}(x),$$

where $\mathrm{SC}(x)$ is in general a proper subset of $\mathrm{RSSS}(x)$.

Proof. The inclusions $\mathrm{USS}(x) \subseteq \mathrm{SSS}(x) \subseteq \mathrm{SS}(x)$ hold by definition. To show the inclusion $\mathrm{RSSS}(x) \subseteq \mathrm{USS}(x)$ one just needs to prove that $\mathrm{RSSS}(x) \subseteq \mathrm{SSS}(x)$. This follows from [15], where it is shown that iterated cycling increases the infimum of an element until the maximum of the infimum in the conjugacy class is reached, and that iterated decycling decreases the supremum of an element until the minimum of the supremum in the conjugacy class is reached.

It is also clear from the definitions that if the elements in $\mathrm{SSS}(x)$ have canonical length 1, then $\mathrm{RSSS}(x) = \mathrm{USS}(x) = \mathrm{SSS}(x)$, since cycling and decycling restrict to the finite order maps τ^{-p} and τ^p when applied to elements of canonical length 1.

Hence it just remains to be shown that $\mathrm{SC}(x) \subseteq \mathrm{RSSS}(x)$, and that equality holds if the canonical length of its elements is greater than one. By Corollary 3.9, iterated cyclic sliding decreases the canonical length of an element to its minimum in the conjugacy class, hence $\mathrm{SC}(x) \subseteq \mathrm{SSS}(x)$. Suppose first that $\ell_s(x) > 1$, so one can apply Lemma 3.8. In this case every element $z \in \mathrm{SSS}(x)$ falls within Case 4 in Lemma 3.8, that is, $\mathfrak{s}(z) = \mathbf{c}(\mathbf{d}(z)) = \mathbf{d}(\mathbf{c}(z))$. In particular, cycling and decycling commute on $\mathrm{SSS}(x)$ and we have $\mathfrak{s}^m(z) = \mathbf{c}^m(\mathbf{d}^m(z)) = \mathbf{d}^m(\mathbf{c}^m(z))$ for every $z \in \mathrm{SSS}(x)$ and every $m \geq 1$. Since $\mathrm{SSS}(x)$ is a finite set, closed under cycling and decycling, there is a common upper bound N such that $\mathbf{c}^n(z)$ belongs to a circuit under cycling and $\mathbf{d}^n(z)$ belongs to a circuit under decycling for every $z \in \mathrm{SSS}(x)$ and every $n \geq N$. Now let $y \in \mathrm{SC}(x)$ and assume that N is a multiple of the length of the period of y under sliding, that is, $\mathfrak{s}^N(y) = y$. Then one has that $y = \mathfrak{s}^N(y) = \mathbf{c}^N(\mathbf{d}^N(y))$ belongs to a circuit under cycling and at the same time that $y = \mathfrak{s}^N(y) = \mathbf{d}^N(\mathbf{c}^N(y))$ belongs to a circuit under decycling. Hence $y \in \mathrm{RSSS}(x)$. Conversely, if $y \in \mathrm{RSSS}(x)$ and $\ell(y) > 1$, then we consider M such that $\mathbf{c}^M(y) = y$ and also $\mathbf{d}^M(y) = y$. Then one has $\mathfrak{s}^M(y) = \mathbf{d}^M(\mathbf{c}^M(y)) = \mathbf{d}^M(y) = y$, so $y \in \mathrm{SC}(x)$.

Finally, since $\mathrm{SC}(x) \subseteq \mathrm{SSS}(x)$ in any case, and $\mathrm{RSSS}(x) = \mathrm{SSS}(x)$ if their elements have canonical length 1, it follows that $\mathrm{SC}(x) \subseteq \mathrm{RSSS}(x)$ in any case. If $\ell_s(x) = 1$, the equality does not hold in general, as one can see in the following example in the Artin braid group on 4 strands. Let $x = \sigma_1\sigma_2\sigma_3 \in B_4$. Then $\mathrm{RSSS}(x) = \mathrm{USS}(x) = \mathrm{SSS}(x) = \{\sigma_1\sigma_2\sigma_3, \sigma_3\sigma_2\sigma_1, \sigma_2\sigma_1\sigma_3, \sigma_1\sigma_3\sigma_2\}$, but one has $\mathrm{SC}(x) = \{\sigma_2\sigma_1\sigma_3, \sigma_1\sigma_3\sigma_2\}$, since $\mathfrak{s}(\sigma_1\sigma_2\sigma_3) = \mathfrak{s}(\sigma_3\sigma_2\sigma_1) = \mathfrak{s}(\sigma_1\sigma_3\sigma_2) = \sigma_1\sigma_3\sigma_2$ and $\mathfrak{s}(\sigma_2\sigma_1\sigma_3) = \sigma_2\sigma_1\sigma_3$. □

As a conclusion, the set $\mathrm{SC}(x)$ that we introduced in this paper is a (in general proper) subset of the sets that were defined similarly in previous papers. Although $\mathrm{SC}(x)$ is equal to $\mathrm{RSSS}(x)$ in most cases, the case $\ell_s(x) = 1$ in which the sets differ is not irrelevant. For instance, in the braid group B_n, a periodic braid x which is not conjugate to a power of Δ has summit length 1, but the conjugacy problem for such braids is far from being an easy issue [7].

We defined the set of sliding circuits $\mathrm{SC}(x)$ as a subset of x^G above. However, in order to be able to compute it algorithmically, and in particular to use it for solving the conjugacy search problem, we need to know how its elements are related by conjugations. One particularly simple way to achieve this is by means of a directed graph; this is the basis of the algorithms in [18, 15, 8, 17, 19].

For this purpose, it will be convenient to display conjugations in a graph-theoretical style: we shall write $u \xrightarrow{s} v$ if $u^s = v$ for some $u, s, v \in G$. Hence we have, for instance:

$$x \xrightarrow{\mathfrak{p}(x)} \mathfrak{s}(x).$$

Then we can define, given $x \in G$, a directed graph whose vertices correspond to the elements of $\mathrm{SC}(x)$ and whose arrows correspond to certain conjugating elements, each sending one particular element in $\mathrm{SC}(x)$ to another. We will define this graph and analyse its properties in §3.4. Before getting to that, however, we need to describe an important map that transforms conjugating elements, the *transport map.*

3.3 The transport map

Given two conjugate elements x and $x^\alpha = \alpha^{-1}x\alpha$, the images of x and x^α under cyclic sliding are also conjugate and we will frequently want to relate α to a conjugating element for the images $\mathfrak{s}(x)$ and $\mathfrak{s}(x^\alpha)$. This can be done using the notion of *transport*:

Definition 3.14. *Given $x, \alpha \in G$, we define the* **transport** *of α at x under cyclic sliding as*

$$\alpha^{(1)} = \mathfrak{p}(x)^{-1}\, \alpha\, \mathfrak{p}(x^\alpha).$$

That is, $\alpha^{(1)}$ is the conjugating element that makes the following diagram commutative, in the sense that the conjugating element along any closed path is trivial:

$$\begin{tikzcd}
x \arrow[r, "\mathfrak{p}(x)"] \arrow[d, "\alpha"'] & \mathfrak{s}(x) \arrow[d, "\alpha^{(1)}"] \\
x^\alpha \arrow[r, "\mathfrak{p}(x^\alpha)"'] & \mathfrak{s}(x^\alpha)
\end{tikzcd}$$

Note that the horizontal rows in this diagram correspond to applications of cyclic sliding.

For an integer $i > 1$ we define recursively $\alpha^{(i)} = (\alpha^{(i-1)})^{(1)}$. Note that $(\alpha^{(i-1)})^{(1)}$ indicates the transport of $\alpha^{(i-1)}$ at $\mathfrak{s}^{i-1}(x)$. We also define $\alpha^{(0)} = \alpha$.

There is an interpretation of the transport under cyclic sliding in terms of category theory. We can consider G as a category, in which the objects are the elements of G and the morphisms correspond to conjugations, as in the above diagram. Then cyclic sliding can be seen as a functor from G to itself, sending an object x to $\mathfrak{s}(x)$, and a morphism α from x to y, to the morphism $\alpha^{(1)}$ from $\mathfrak{s}(x)$ to $\mathfrak{s}(y)$. That is, the transport is the natural way to define the image of a morphism under the functor $\mathfrak{s}$. Notice that $\mathfrak{s}$ can also be considered as a functor from $\mathrm{SSS}(x)$ (respectively $\mathrm{USS}(x)$, $\mathrm{RSSS}(x)$ or $\mathrm{SC}(x)$) to itself. Moreover, the functor $\mathfrak{s}$ is an isomorphism of categories when restricted to $\mathrm{SC}(x)$.

3.3.1 Properties of the transport

Under certain conditions, the transport under cyclic sliding respects many aspects of the Garside structure. In particular, we will see that if x and x^α as above are super summit elements, the transport respects products, left divisibility and gcds and leaves powers of Δ invariant.

Lemma 3.15. *Let $x, \alpha \in G$ such that $\inf(x^\alpha) \leq \inf(x)$ and $\sup(x^\alpha) \geq \sup(x)$ and consider the transport $\alpha^{(1)}$ of α at x. If α is positive then $\alpha^{(1)}$ is positive.*

Proof. Since $\alpha^{(1)} = \mathfrak{p}(x)^{-1}\, \alpha\, \mathfrak{p}(x^\alpha)$, we must show that $\mathfrak{p}(x) \preccurlyeq \alpha\, \mathfrak{p}(x^\alpha)$. Let $y = x^\alpha$. We can write $\iota(y) = y\Delta^{-\inf(y)} \wedge \Delta = \alpha^{-1}x\alpha\Delta^{-\inf(y)} \wedge \Delta$, that is, $\alpha\, \iota(y) = x\alpha\Delta^{-\inf(y)} \wedge \alpha\Delta$. Since $\inf(y) \leq \inf(x)$ and α is positive, we have $x\Delta^{-\inf(x)} \preccurlyeq x\alpha\Delta^{-\inf(y)}$ and also $\Delta \preccurlyeq \alpha\Delta$, whence we obtain $\iota(x) = x\Delta^{-\inf(x)} \wedge \Delta \preccurlyeq x\alpha\Delta^{-\inf(y)} \wedge \alpha\Delta = \alpha\iota(y)$. Analogously, as $\sup(y) \geq \sup(x)$ is equivalent to $\inf(y^{-1}) \leq \inf(x^{-1})$, we also have $\iota(x^{-1}) \preccurlyeq \alpha\iota(y^{-1})$. Together, these imply $\mathfrak{p}(x) = \iota(x) \wedge \iota(x^{-1}) \preccurlyeq \alpha\iota(y) \wedge \alpha\iota(y^{-1}) = \alpha\left(\iota(y) \wedge \iota(y^{-1})\right) = \alpha\mathfrak{p}(y)$ as claimed. □

Lemma 3.16. *Let $x, \alpha \in G$ and consider the transport $\alpha^{(1)}$ of α at x. If $\alpha = \Delta^k$ for $k \in \mathbb{Z}$ then $\alpha^{(1)} = \Delta^k$.*

$$\begin{tikzcd}
x \arrow[r, "\mathfrak{p}(x)"] \arrow[d, "\Delta^k"'] & \mathfrak{s}(x) \arrow[d, "\Delta^k"] \\
\tau^k(x) \arrow[r, "\mathfrak{p}(\tau^k(x))"'] & \mathfrak{s}(\tau^k(x))
\end{tikzcd}$$

Proof. We have $x^\alpha = \tau^k(x)$ and $\mathfrak{p}(x^\alpha) = \tau^k(\mathfrak{p}(x))$, whence $\alpha^{(1)} = \mathfrak{p}(x)^{-1}\Delta^k\tau^k(\mathfrak{p}(x)) = \Delta^k$. □

Lemma 3.17. *Let $x, \alpha, \beta \in G$ and consider the transports $\alpha^{(1)}$ of α and $(\alpha\beta)^{(1)}$ of $\alpha\beta$ at x and the transport $\beta^{(1)}$ of β at x^α. Then $(\alpha\beta)^{(1)} = \alpha^{(1)}\beta^{(1)}$.*

$$\begin{tikzcd}
x \arrow[r, "\mathfrak{p}(x)"] \arrow[d, "\alpha"'] & \mathfrak{s}(x) \arrow[d, "\alpha^{(1)}"] \\
x^\alpha \arrow[r, "\mathfrak{p}(x^\alpha)"] \arrow[d, "\beta"'] & \mathfrak{s}(x^\alpha) \arrow[d, "\beta^{(1)}"] \\
x^{\alpha\beta} \arrow[r, "\mathfrak{p}(x^{\alpha\beta})"] & \mathfrak{s}(x^{\alpha\beta})
\end{tikzcd}$$

Proof. Trivial, by construction. □

Corollary 3.18. *Let $x, \alpha, \gamma \in G$ such that $\inf(x^\gamma) \leq \inf(x^\alpha)$ and $\sup(x^\gamma) \geq \sup(x^\alpha)$ and consider the transports $\alpha^{(1)}$ of α and $\gamma^{(1)}$ of γ at x. Then if $\alpha \preccurlyeq \gamma$, one has $\alpha^{(1)} \preccurlyeq \gamma^{(1)}$.*

Proof. Recall that $\alpha \preccurlyeq \gamma$ if and only if $\beta = \alpha^{-1}\gamma$ is positive. As $\gamma = \alpha\beta$, we have $\gamma^{(1)} = \alpha^{(1)}\beta^{(1)}$ by Lemma 3.17. Since $\beta^{(1)}$ is positive by Lemma 3.15, this implies $\alpha^{(1)} \preccurlyeq \gamma^{(1)}$. □

Corollary 3.19. *Let $x, \alpha \in G$ such that $\inf(x^\alpha) = \inf(x)$ and $\sup(x^\alpha) = \sup(x)$ and consider the transport $\alpha^{(1)}$ of α at x. Then $\inf(\alpha^{(1)}) \geq \inf(\alpha)$ and $\sup(\alpha^{(1)}) \leq \sup(\alpha)$, hence $\ell(\alpha^{(1)}) \leq \ell(\alpha)$. In particular, if α is simple then so is $\alpha^{(1)}$.*

Proof. Let $p = \inf(\alpha)$ and $q = \sup(\alpha)$. Firstly note that $\inf(x^{\Delta^p}) = \inf(x) = \inf(x^{\Delta^q})$ and $\sup(x^{\Delta^p}) = \sup(x) = \sup(x^{\Delta^q})$. By Lemma 3.16 and Corollary 3.18, $\Delta^p \preccurlyeq \alpha \preccurlyeq \Delta^q$ then implies $\Delta^p \preccurlyeq \alpha^{(1)} \preccurlyeq \Delta^q$. □

Proposition 3.20. *Let $x, \alpha, \beta \in G$ such that $\inf(x^\alpha) = \inf(x^{\alpha\wedge\beta}) = \inf(x^\beta)$ and $\sup(x^\alpha) = \sup(x^{\alpha\wedge\beta}) = \sup(x^\beta)$ and consider the transports $\alpha^{(1)}$ of α, $\beta^{(1)}$ of β and $(\alpha\wedge\beta)^{(1)}$ of $\alpha\wedge\beta$ at x. Then $(\alpha\wedge\beta)^{(1)} = \alpha^{(1)} \wedge \beta^{(1)}$.*

Proof. By replacing x by $x^{\alpha\wedge\beta}$ and using Lemma 3.17, we can assume $\alpha \wedge \beta = 1$.

Denoting $p = \inf(x) = \inf(x^\alpha) = \inf(x^\beta)$, we can write $\iota(x^\alpha) = x^\alpha \Delta^{-p} \wedge \Delta = \alpha^{-1}x\alpha\Delta^{-p} \wedge \Delta$, whence $\alpha\iota(x^\alpha) = x\Delta^{-p}\tau^{-p}(\alpha) \wedge \Delta\tau(\alpha)$. Similarly, $\beta\iota(x^\beta) = x\Delta^{-p}\tau^{-p}(\beta) \wedge \Delta\tau(\beta)$. Together, these imply

$$\alpha\iota(x^\alpha) \wedge \beta\iota(x^\beta) \;=\; x\Delta^{-p}\tau^{-p}(\alpha\wedge\beta) \wedge \Delta\tau(\alpha\wedge\beta) \;=\; x\Delta^{-p} \wedge \Delta \;=\; \iota(x).$$

Analogously, $\alpha\iota((x^\alpha)^{-1}) \wedge \beta\iota((x^\beta)^{-1}) = \iota(x^{-1})$. We hence obtain

$$\begin{aligned}
\alpha^{(1)} \wedge \beta^{(1)} &= \big(\iota(x) \wedge \iota(x^{-1})\big)^{-1} \alpha \left(\iota(x^\alpha) \wedge \iota((x^\alpha)^{-1})\right) \\
&\qquad \wedge \big(\iota(x) \wedge \iota(x^{-1})\big)^{-1} \beta \left(\iota(x^\beta) \wedge \iota((x^\beta)^{-1})\right) \\
&= \big(\iota(x) \wedge \iota(x^{-1})\big)^{-1} \left(\alpha\iota(x^\alpha) \wedge \beta\iota(x^\beta) \wedge \alpha\iota((x^\alpha)^{-1}) \wedge \beta\iota((x^\beta)^{-1})\right) \\
&= \big(\iota(x) \wedge \iota(x^{-1})\big)^{-1} \left(\iota(x) \wedge \iota(x^{-1})\right) \;=\; 1
\end{aligned}$$

as claimed. □

3.3.2 Right transport and the reverse Garside structure

Recall that in a Garside group (G, P, Δ), apart from the prefix order $\preccurlyeq$, one also has the suffix order $\succcurlyeq$, defined by $a \succcurlyeq b$ if and only if $ab^{-1} \in P$. With respect to the latter, one can consider the the notions of preferred suffix and cyclic right sliding, which are analogous to the preferred prefix and cyclic sliding, but refer to the partial order $\succcurlyeq$ instead of $\preccurlyeq$.

Definition 3.21. *Given $x \in G$, we define the* **preferred suffix** $\mathfrak{p}^{\Lsh}(x)$ *of x as the simple element*

$$\mathfrak{p}^{\Lsh}(x) = \left(\Delta^{-\inf(x)}x\right) \wedge^{\Lsh} \left(\Delta^{\sup(x)}x^{-1}\right) \wedge^{\Lsh} \Delta.$$

Definition 3.22. *Given $x \in G$, we define the* **cyclic right sliding** $\mathfrak{s}^{\Lsh}(x)$ *of x as the conjugate of x by the inverse of its preferred suffix:*

$$\mathfrak{s}^{\Lsh}(x) = x^{\mathfrak{p}^{\Lsh}(x)^{-1}} = \mathfrak{p}^{\Lsh}(x)\; x\; \mathfrak{p}^{\Lsh}(x)^{-1}.$$

This implies that one can also define a transport map for cyclic right sliding, as follows. We remark that, when one considers these notions with respect to $\succcurlyeq$, and tries to relate them to the analogous notions with respect to $\preccurlyeq$, one must consider conjugating elements *on the left*, meaning that a (left) conjugating element α relates x to $x^{\alpha^{-1}} = \alpha x \alpha^{-1}$.

Definition 3.23. *Given $x, \alpha \in G$, we define the* **right transport** *of α at x under cyclic right sliding as $\alpha^{(1)^{\Lsh}} = \mathfrak{p}^{\Lsh}(x^{\alpha^{-1}})\, \alpha\, \mathfrak{p}^{\Lsh}(x)^{-1}$. That is, $\alpha^{(1)^{\Lsh}}$ is the conjugating element that makes the following diagram commutative, in the sense that the conjugating element along any closed path is trivial:*

$$\begin{tikzcd}
x & \mathfrak{s}^{\Lsh}(x) \arrow[l, "\mathfrak{p}^{\Lsh}(x)"'] \\
x^{\alpha^{-1}} \arrow[u, "\alpha"] & \mathfrak{s}^{\Lsh}(x^{\alpha^{-1}}) \arrow[l, "\mathfrak{p}^{\Lsh}(x^{\alpha^{-1}})"] \arrow[u, "\alpha^{(1)^{\Lsh}}"']
\end{tikzcd}$$

For an integer $i > 1$ we define recursively $\alpha^{(i)^{\Lsh}} = (\alpha^{(i-1)^{\Lsh}})^{(1)^{\Lsh}}$. Note that $(\alpha^{(i-1)^{\Lsh}})^{(1)^{\Lsh}}$ indicates the right transport of $\alpha^{(i-1)^{\Lsh}}$ at ${\mathfrak{s}^{\Lsh}}^{i-1}(x)$. We also define $\alpha^{(0)^{\Lsh}} = \alpha$.

All results obtained for cyclic (left) sliding and (left) transport in this section hold in analogous form for cyclic right sliding and right transport; the proofs can be translated in a straight-forward way. However, instead of duplicating all the proofs, we will consider a different Garside structure of G, which is related to the Garside structure (G, P, Δ) for G fixed earlier.

Proposition 3.24. *1. The triple (G, P^{-1}, Δ^{-1}) is also a Garside structure of G, which we refer to as the* **reverse Garside structure**. *We denote the associated partial orderings by $\preccurlyeq_*$ respectively $\succcurlyeq_*$, the lattice operations by $\wedge_*$, $\vee_*$, $\wedge^{\Lsh}_*$ and $\vee^{\Lsh}_*$, and infimum, supremum and canonical length with respect to (G, P^{-1}, Δ^{-1}) by $\inf_*$, $\sup_*$ and ℓ_*.*

2. For $a, b \in G$, the following are equivalent: (a) $a \preccurlyeq b$ (b) $a^{-1} \succcurlyeq b^{-1}$ (c) $b \preccurlyeq_* a$ (d) $b^{-1} \succcurlyeq_* a^{-1}$

3. For any $x \in G$, one has $\inf_*(x) = -\sup(x)$, $\sup_*(x) = -\inf(x)$ and $\ell_*(x) = \ell(x)$. In particular, x is super summit with respect to (G, P, Δ) if and only if x is super summit with respect to (G, P^{-1}, Δ^{-1}).

4. For $a, b, c \in G$, the following are equivalent: (a) $a = b \wedge c$ (b) $a^{-1} = b^{-1} \vee^{\Lsh} c^{-1}$ (c) $a = b \vee_* c$ (d) $a^{-1} = b^{-1} \wedge^{\Lsh}_* c^{-1}$

5. For $a, b, c \in G$, the following are equivalent: (a) $a = b \vee c$ (b) $a^{-1} = b^{-1} \wedge^{\Lsh} c^{-1}$ (c) $a = b \wedge_* c$ (d) $a^{-1} = b^{-1} \vee^{\Lsh}_* c^{-1}$

Proof. Since $a^{-1}b = a^{-1}(b^{-1})^{-1} = (b^{-1}a)^{-1} = (b^{-1}(a^{-1})^{-1})^{-1}$, all statements in Claim 2 are equivalent to $a^{-1}b \in P$.

For Claim 4 note that $a = b \wedge c$ means $(a \preccurlyeq b) \wedge (a \preccurlyeq c) \wedge \big(\forall d : (d \preccurlyeq a) \vee \neg(d \preccurlyeq b) \vee \neg(d \preccurlyeq c)\big)$. By Claim 2, the latter is equivalent to $(b \preccurlyeq_* a) \wedge (c \preccurlyeq_* a) \wedge \big(\forall d : (a \preccurlyeq_* d) \vee \neg(b \preccurlyeq_* d) \vee \neg(c \preccurlyeq_* d)\big)$, that is, to $a = b \vee_* c$. The other equivalences in Claims 4 and 5 can be proved in the same way.

Again by Claim 2, $\Delta^p \preccurlyeq x \preccurlyeq \Delta^q$ is equivalent to $(\Delta^{-1})^{-q} \preccurlyeq_* x \preccurlyeq_* (\Delta^{-1})^{-p}$, showing Claim 3.

In particular, $\preccurlyeq_*$ and $\succcurlyeq_*$ are lattice orders, and since the set $[1, \Delta]$ generates G, so does the set $[1, \Delta^{-1}]_* = \{a \in G \,|\, 1 \preccurlyeq_* a \preccurlyeq_* \Delta^{-1}\} = \{a^{-1} \in G \,|\, a \in [1, \Delta]\}$. Moreover, as $\Delta^{-1}P\Delta = P$, we have $(\Delta^{-1})^{-1}P^{-1}\Delta^{-1} = P^{-1}$. Finally, if $x \in P^{-1}\backslash\{1\}$ and $x = a_1 \cdots a_k$ where $a_i \in P^{-1}\backslash\{1\}$ for $i = 1, \ldots, k$, then $x^{-1} = a_k^{-1} \cdots a_1^{-1} \in P\backslash\{1\}$ and $a_k^{-1} \in P\backslash\{1\}$ for $i = 1, \ldots, k$. Hence,

$$||x||_* := \sup\big\{k \mid \exists\, a_1, \ldots, a_k \in P^{-1}\backslash\{1\} \text{ such that } x = a_1 \cdots a_k\big\} \leq ||x^{-1}|| < \infty.$$

Thus, (G, P^{-1}, Δ^{-1}) is a Garside structure of G and Claim 1 is shown. □

Corollary 3.25. *For $x \in G$ we denote by $\mathfrak{p}_*(x)$ and $\mathfrak{s}_*(x)$ the preferred prefix of x respectively the cyclic (left) sliding of x with respect to the Garside structure (G, P^{-1}, Δ^{-1}). Then,*

$$\mathfrak{p}_*(x) = \mathfrak{p}^{\Lsh}(x)^{-1} \quad \textit{and} \quad \mathfrak{s}_*(x) = \mathfrak{s}^{\Lsh}(x).$$

Proof. By the definitions of $\mathfrak{p}_*(x)$ and $\mathfrak{p}^{\Lsh}(x)$ and Proposition 3.24 (3) and (5) we have

$$\begin{aligned} \mathfrak{p}_*(x) &= x(\Delta^{-1})^{-\inf_*(x)} \wedge_* x^{-1}(\Delta^{-1})^{\sup_*(x)} \wedge_* \Delta^{-1} \\ &= \Big(\Delta^{\sup(x)}x^{-1} \wedge^{\Lsh} \Delta^{-\inf(x)}x \wedge^{\Lsh} \Delta\Big)^{-1} \;=\; \mathfrak{p}^{\Lsh}(x)^{-1}. \end{aligned}$$

In particular, $\mathfrak{s}^{\Lsh}(x) = x^{\mathfrak{p}^{\Lsh}(x)^{-1}} = x^{\mathfrak{p}_*(x)} = \mathfrak{s}_*(x)$. □

Hence, cyclic right sliding and right transport with respect to the Garside structure (G, P, Δ) are equivalent to cyclic (left) sliding and (left) transport with respect to the reverse Garside structure (G, P^{-1}, Δ^{-1}). In particular, all results for cyclic (left) sliding and (left) transport can be translated to the corresponding results for cyclic right sliding and right transport. Note that, in doing so, $\preccurlyeq$ is replaced by $\succcurlyeq$ (and hence $\wedge$ and $\vee$ by $\wedge^{\Lsh}$ and $\vee^{\Lsh}$, respectively) and usual (right) conjugation is replaced by *left* conjugation (where the left conjugate of x by c is $c \cdot x \cdot c^{-1} = x^{c^{-1}}$).

We finish with a result relating cyclic (left) sliding and cyclic right sliding.

Proposition 3.26. *Let $x \in G$. Then for any $z \in \mathrm{SSS}(x)$ one has $\mathfrak{p}_*(\mathfrak{s}(z))^{-1} = \mathfrak{p}^{\Lsh}(\mathfrak{s}(z)) \succcurlyeq \mathfrak{p}(z)$ and $\mathfrak{p}_*(z)^{-1} = \mathfrak{p}^{\Lsh}(z) \preccurlyeq \mathfrak{p}(\mathfrak{s}^{\Lsh}(z)) = \mathfrak{p}(\mathfrak{s}_*(z))$. In particular, $\mathfrak{p}(z) \cdot \mathfrak{p}_*(\mathfrak{s}(z)) \in P^{-1}$ and $\mathfrak{p}_*(z) \cdot \mathfrak{p}(\mathfrak{s}_*(z)) \in P$.*

Proof. Let $p = \inf(z)$ and $q = \sup(z)$. By the definition of $\mathfrak{p}(z)$ we have $\mathfrak{p}(z) \preccurlyeq z\Delta^{-p}$ and $\mathfrak{p}(z) \preccurlyeq z^{-1}\Delta^{q}$. Hence, $\Delta^{-p}\mathfrak{p}(z)^{-1}z$ and $\Delta^{q}\mathfrak{p}(z)^{-1}z^{-1}$ are positive elements and we obtain

$$\begin{aligned} \mathfrak{p}_*(\mathfrak{s}(z))^{-1} &= \mathfrak{p}^{\Lsh}(\mathfrak{s}(z)) = \Delta^{-p}\mathfrak{s}(z) \wedge^{\Lsh} \Delta^{q}\mathfrak{s}(z)^{-1} \wedge^{\Lsh} \Delta \\ &= \Delta^{-p}\mathfrak{p}(z)^{-1}z\mathfrak{p}(z) \wedge^{\Lsh} \Delta^{q}\mathfrak{p}(z)^{-1}z^{-1}\mathfrak{p}(z) \wedge^{\Lsh} \Delta \succcurlyeq \mathfrak{p}(z) \end{aligned}$$

using Corollary 3.25 and noting that $\mathfrak{p}(z)$ is simple. Applying the same argument to the reverse Garside structure, we also have $\mathfrak{p}(\mathfrak{s}_*(z))^{-1} \succcurlyeq_* \mathfrak{p}_*(z)$, which by Proposition 3.24 is equivalent to $\mathfrak{p}_*(z)^{-1} \preccurlyeq \mathfrak{p}(\mathfrak{s}_*(z))$. Finally, Corollary 3.25 yields $\mathfrak{p}_*(z)^{-1} = \mathfrak{p}^{\Lsh}(z)$ and $\mathfrak{p}(\mathfrak{s}^{\Lsh}(z)) = \mathfrak{p}(\mathfrak{s}_*(z))$. □

3.4 Sliding circuits graph

We are now able to define a directed graph structure on the set of sliding circuits, which we can use to solve the conjugacy problems in G, and to analyse its properties. The main result of this section is Corollary 3.35, which shows that the resulting graph is finite and connected.

We will define a graph $\mathrm{SCG}(x)$, whose vertices are the elements of $\mathrm{SC}(x)$ and whose arrows correspond to conjugating elements. In order to be able to compute the graph, we need it to be connected. We could proceed as in [15], showing that the elements of $\mathrm{SC}(x)$ are connected through conjugations by simple elements and using all simple elements which conjugate a given element $y \in \mathrm{SC}(x)$ to another element of $\mathrm{SC}(x)$ as the arrows starting at the vertex y. However, applying an improvement from [17] substantially reduces the number of arrows. The idea is to fix a vertex $y \in \mathrm{SC}(x)$ of the graph, consider the set of positive elements of G that conjugate y to another element of $\mathrm{SC}(x)$, and to define the arrows of $\mathrm{SCG}(x)$ starting at y to be the *minimal* elements (with respect to $\preccurlyeq$) in this set of conjugating elements. We shall see in Corollary 3.35 that the arrows defined in this way are simple elements.

Definition 3.27. *Given $x \in G$ and $y \in \mathrm{SC}(x)$, we say that a positive element $s \in P \setminus \{1\}$ is an* **indecomposable conjugator starting at** *y, if $y^s \in \mathrm{SC}(x)$ and it is not possible to decompose s as a product of two nontrivial positive elements $s = s_1s_2$ ($s_1, s_2 \neq 1$), in such a way that $y^{s_1} \in \mathrm{SC}(x)$. In other words, s is an indecomposable conjugator starting at y, if no nontrivial prefix of s conjugates y to an element in $\mathrm{SC}(x)$.*

Definition 3.28. *Given $x \in G$, we define $\mathrm{SCG}(x)$, the* **sliding circuits graph** *of x, to be the directed graph whose vertices are the elements of $\mathrm{SC}(x)$ and whose arrows are the indecomposable conjugators starting at y for every vertex $y \in \mathrm{SC}(x)$.*

We now show that $\mathrm{SCG}(x)$ is finite and connected, following the arguments in [17] and [19].

Proposition 3.29. *Let $x \in G$. If $x^\alpha, x^\beta \in \mathrm{SSS}(x)$ for elements $\alpha, \beta \in G$, then $x^{\alpha\wedge\beta} \in \mathrm{SSS}(x)$.*

Proof. Let $t = \alpha \wedge \beta$ and write $\alpha = t\overline{\alpha}$ and $\beta = t\overline{\beta}$. Notice that $\overline{\alpha}$ and $\overline{\beta}$ are positive and $\overline{\alpha} \wedge \overline{\beta} = 1$. Let $p = \inf_s(x) = \inf(x^\alpha) = \inf(x^\beta)$. Then $\Delta^p \preccurlyeq \overline{\alpha}x^\alpha = x^t\overline{\alpha}$ and $\Delta^p \preccurlyeq \overline{\beta}x^\beta = x^t\overline{\beta}$, that is, $\Delta^p \preccurlyeq x^t(\overline{\alpha}\wedge\overline{\beta}) = x^t$. As p is the maximal infimum of conjugates of x, we have $\inf(x^t) = p$. Applying the same argument to the inverses of x^t, x^α and x^β and observing $\sup(z) = -\inf(z^{-1})$ for $z \in G$ we obtain $\sup(x^t) = \sup_s(x)$, that is, $x^t \in \mathrm{SSS}(x)$. □

Corollary 3.30. *Let $x \in G$. If $x^\alpha, x^\beta \in \mathrm{SSS}(x)$ for elements $\alpha, \beta \in G$, then $x^{\alpha\vee\beta} \in \mathrm{SSS}(x)$.*

Proof. Let $\mathrm{SSS}_*(x)$ denote the super summit set of x with respect to the reverse Garside structure (G, P^{-1}, Δ^{-1}). By Proposition 3.24 (3), we have $\mathrm{SSS}_*(x) = \mathrm{SSS}(x)$. Using Proposition 3.24 (5) and Proposition 3.29, $x^\alpha, x^\beta \in \mathrm{SSS}(x) = \mathrm{SSS}_*(x)$ yields $x^{\alpha\vee\beta} = x^{\alpha\wedge_*\beta} \in \mathrm{SSS}_*(x) = \mathrm{SSS}(x)$. □

Corollary 3.31. *Let $x \in G$. There is a unique positive element $\rho(x)$ (possibly trivial) satisfying the following.*

1. *$x^{\rho(x)} \in \mathrm{SSS}(x)$.*
2. *$\rho(x) \preccurlyeq \alpha$ for every positive $\alpha \in G$ satisfying $x^\alpha \in \mathrm{SSS}(x)$.*

Proof. As some power Δ^e of Δ is central in G, we can choose a positive element $c \in G$ such that $x^c \in \mathrm{SSS}(x)$. Now consider the set $D = \{\alpha \in G \mid 1 \preccurlyeq \alpha \preccurlyeq c \text{ and } x^\alpha \in \mathrm{SSS}(x)\}$. As $\sup(\alpha) \leq \sup(c)$ for all $\alpha \in D$ and G is of finite type, the set D is finite. Moreover, D is non-empty as $c \in D$. Hence we can define $\rho = \bigwedge_{\alpha\in D} \alpha$.

The element ρ is positive and we have $x^\rho \in \mathrm{SSS}(x)$ by Proposition 3.29. Moreover, for any $\alpha \in G$ satisfying $1 \preccurlyeq \alpha$ and $x^\alpha \in \mathrm{SSS}(x)$ we have $\alpha \wedge c \in D$, again by Proposition 3.29, and hence $\rho \preccurlyeq \alpha \wedge c \preccurlyeq \alpha$, that is, ρ has the required properties. If ρ' is another element with the required properties, one has $\rho \preccurlyeq \rho'$ and $\rho' \preccurlyeq \rho$, so ρ is unique. □

The computation of the element $\rho(x)$ is given in [17], an alternative simpler way can be found in [20]. Notice that, in the above situation, if $x \in \mathrm{SSS}(x)$ then $\rho(x) = 1$.

Lemma 3.32. *Let $x \in G$, $y \in \mathrm{SC}(x)$ and $s \in G$ such that $y^s \in \mathrm{SSS}(x)$. Let N be a positive integer such that $\mathfrak{s}^N(y) = y$ and for integers $i \geq 0$ consider the transports $s^{(iN)}$ at y. Then the following hold.*

1. *There are integers $0 \leq i_1 < i_2$ such that $s^{(i_1N)} = s^{(i_2N)}$.*
2. *$y^s \in \mathrm{SC}(x)$ if and only if there is a positive integer k such that $s^{(kN)} = s$.*

Proof. As $\mathrm{SC}(x) \subseteq \mathrm{SSS}(x)$, Corollary 3.19 yields $\inf(s^{(iN)}) \geq \inf(s)$ and $\sup(s^{(iN)}) \leq \sup(s)$ for all $i \in \mathbb{N}$. As G is of finite type, the set of elements with given infimum and canonical length is finite, whence there must be $i_1 < i_2 \in \mathbb{N}$ such that $s^{(i_1N)} = s^{(i_2N)}$, proving the first claim.

To show the second claim, assume first that $y^s \in \mathrm{SC}(x)$. Replacing N by a multiple, if necessary, we can assume that $\mathfrak{s}^N(y^s) = y^s$. Denote by $\mathcal{C}$ the set of elements conjugating y to y^s. Then, denoting $\alpha = \mathfrak{p}(y)\cdots\mathfrak{p}(\mathfrak{s}^{N-1}(y))$ and $\beta = \mathfrak{p}(y^s)\cdots\mathfrak{p}(\mathfrak{s}^{N-1}(y^s))$, we have $t^{(N)} = \alpha^{-1}\cdot t\cdot\beta$ for every $t \in \mathcal{C}$, that is, the map $\varphi : \mathcal{C} \to \mathcal{C}$ that sends t to $t^{(N)}$ is bijective. Together with Claim 1 this implies the existence of $i \in \mathbb{N}$ such that $s^{(iN)} = s$. Conversely, assume that there is $k > 0$ such that $s^{(kN)} = s$. Then we have, by the definition of the transport, $\mathfrak{s}^{kN}(y^s) = (\mathfrak{s}^{kN}(y))^{(s^{(kN)})} = y^s$, that is, $y^s \in \mathrm{SC}(x)$. □

Proposition 3.33. *Let $x \in G$. If $x^\alpha, x^\beta \in \mathrm{SC}(x)$ for elements $\alpha, \beta \in G$, then $x^{\alpha\wedge\beta} \in \mathrm{SC}(x)$.*

Proof. Let $t = \alpha\wedge\beta$ and write $\alpha = t\overline{\alpha}$ and $\beta = t\overline{\beta}$. Notice that $\overline{\alpha}$ and $\overline{\beta}$ are positive and $\overline{\alpha}\wedge\overline{\beta} = 1$. Since $\mathrm{SC}(x) \subseteq \mathrm{SSS}(x)$, we have $x^t \in \mathrm{SSS}(x)$ by Proposition 3.29. Replacing x by x^t, α by $\overline{\alpha}$ and β by $\overline{\beta}$, we can then assume that α and β are positive, $x \in \mathrm{SSS}(x)$ and $\alpha \wedge \beta = 1$, and we must show that $x \in \mathrm{SC}(x)$. We can moreover assume that x is a minimal counterexample, that is, that $\mathfrak{s}(x) \in \mathrm{SC}(x)$; otherwise apply cyclic sliding to x, x^α and x^β, apply transport to α and β, and note that α and β remain positive by Lemma 3.15 and that the requirement $\alpha \wedge \beta = 1$ is preserved by Proposition 3.20.

Choose $N > 0$ such that $\mathfrak{s}^N(x^\alpha) = x^\alpha$, $\mathfrak{s}^N(x^\beta) = x^\beta$, and $\mathfrak{s}^{N+1}(x) = \mathfrak{s}(x)$ and consider the conjugations indicated in the following commutative diagram; double arrows indicate cyclic sliding.

$$\begin{tikzcd}
x^\alpha \arrow[r, Rightarrow, "\mathfrak{p}(x^\alpha)"] & \mathfrak{s}(x^\alpha) \arrow[r, Rightarrow, "\mathfrak{p}(\mathfrak{s}(x^\alpha))"] & \cdots \arrow[r, Rightarrow] & \mathfrak{s}^N(x^\alpha) = x^\alpha \arrow[r, Rightarrow, "\mathfrak{p}(x^\alpha)"] & \mathfrak{s}(x^\alpha) \\
x \arrow[u, "\alpha"] \arrow[d, "\beta"'] \arrow[r, Rightarrow, "\mathfrak{p}(x)"] & \mathfrak{s}(x) \arrow[u, "\alpha^{(1)}"'] \arrow[d, "\beta^{(1)}"] \arrow[r, Rightarrow, "\mathfrak{p}(\mathfrak{s}(x))"] & \cdots \arrow[r, Rightarrow] & \mathfrak{s}^N(x) \arrow[u, "\alpha^{(N)}"'] \arrow[d, "\beta^{(N)}"] \arrow[r, Rightarrow, "\mathfrak{p}(\mathfrak{s}^N(x))"] & \mathfrak{s}^{N+1}(x) = \mathfrak{s}(x) \arrow[u, "\alpha^{(N+1)}"'] \arrow[d, "\beta^{(N+1)}"] \\
x^\beta \arrow[r, Rightarrow, "\mathfrak{p}(x^\beta)"] & \mathfrak{s}(x^\beta) \arrow[r, Rightarrow, "\mathfrak{p}(\mathfrak{s}(x^\beta))"] & \cdots \arrow[r, Rightarrow] & \mathfrak{s}^N(x^\beta) = x^\beta \arrow[r, Rightarrow, "\mathfrak{p}(x^\beta)"] & \mathfrak{s}(x^\beta)
\end{tikzcd}$$

According to Lemma 3.32, we can assume that $\alpha^{(N+1)} = \alpha^{(1)}$ and $\beta^{(N+1)} = \beta^{(1)}$, replacing N by a suitable multiple if necessary. By Proposition 3.20 we have $\alpha^{(i)} \wedge \beta^{(i)} = 1$ for $i = 1, \ldots, N$ and as all cells in the above diagram commute we obtain

$$\begin{aligned}
\mathfrak{p}(x)^{-1} &= \mathfrak{p}(x)^{-1}(\alpha \wedge \beta) = \mathfrak{p}(x)^{-1}\alpha \wedge \mathfrak{p}(x)^{-1}\beta = \alpha^{(1)}\mathfrak{p}(x^\alpha)^{-1} \wedge \beta^{(1)}\mathfrak{p}(x^\beta)^{-1} \\
&= \alpha^{(N+1)}\mathfrak{p}(x^\alpha)^{-1} \wedge \beta^{(N+1)}\mathfrak{p}(x^\beta)^{-1} = \mathfrak{p}(\mathfrak{s}^N(x))^{-1}\alpha^{(N)} \wedge \mathfrak{p}(\mathfrak{s}^N(x))^{-1}\beta^{(N)} \\
&= \mathfrak{p}(\mathfrak{s}^N(x))^{-1}(\alpha^{(N)} \wedge \beta^{(N)}) = \mathfrak{p}(\mathfrak{s}^N(x))^{-1} .
\end{aligned}$$

Hence, $x = \mathfrak{s}(x)^{\mathfrak{p}(x)^{-1}} = \mathfrak{s}^{N+1}(x)^{\mathfrak{p}(\mathfrak{s}^N(x))^{-1}} = \mathfrak{s}^N(x)$ which implies $x \in \mathrm{SC}(x)$ in contradiction to the choice of x. Hence the claim is shown. □

Corollary 3.34. *Let $x \in G$. There is a unique positive element $c(x)$ (possibly trivial) satisfying the following.*

1. $x^{c(x)} \in \mathrm{SC}(x)$.
2. *$c(x) \preccurlyeq \alpha$ for every positive $\alpha \in G$ satisfying $x^\alpha \in \mathrm{SC}(x)$.*

Proof. The proof is analogous to the proof of Corollary 3.31, with Proposition 3.33 replacing Proposition 3.29. □

Corollary 3.35. *For every $x \in G$, the graph $\mathrm{SCG}(x)$ is finite and connected. Moreover, the arrows of $\mathrm{SCG}(x)$ correspond to simple elements, and the number of arrows starting at a given vertex is bounded above by the number of atoms of G.*

Proof. The elements of $\mathrm{SC}(x)$ have maximal infimum and minimal canonical length in the conjugacy class of x by Lemma 3.4. As G is of finite type, the set of elements with given infimum and canonical length is finite, which implies the finiteness of the set of vertices of $\mathrm{SCG}(x)$. Let $y \in \mathrm{SC}(x)$.

To show that $\mathrm{SCG}(x)$ is connected, suppose $z = y^c \in \mathrm{SC}(x)$. As some power Δ^e of Δ is central in G, we can without loss of generality assume that c is a positive element, replacing c by $\Delta^{me}c$

for suitable m, if necessary. Let $y_1 = y$ and $c_1 = c$. Since $||c||$ is finite, there cannot exist an infinite strictly descending chain of prefixes of c, hence there exists a (not necessarily unique) indecomposable conjugator s_1 starting at y_1 such that $s_1 \preccurlyeq c_1$. If $s_1 \neq c_1$, we can consider $y_2 = y_1^{s_1}$ and $c_2 = s_1^{-1}c_1$: we have $z = y_2^{c_2}$ and can repeat the above argument. Iteratively, we can construct a strictly ascending chain $s_1 \prec s_1 s_2 \prec \ldots \prec s_1 \ldots s_i \preccurlyeq c$. As $||c||$ is finite, this process must terminate, that is, we can decompose $c = s_1 \ldots s_i$ as the product of finitely many indecomposable conjugators starting at $y_1, \ldots, y_i$, respectively, which shows the existence of a path from y to z in $\mathrm{SCG}(x)$.

Now let s be an indecomposable conjugator starting at y. As $y^s \in \mathrm{SC}(x)$ and $\tau(y) = y^{\Delta} \in \mathrm{SC}(x)$, Proposition 3.33 implies $y^{s \wedge \Delta} \in \mathrm{SC}(x)$. If s was not simple, we could write $s = (s \wedge \Delta)t$ for some non-trivial positive element t, contradicting the indecomposability of s.

Finally, Proposition 3.33 implies that for every atom a of G, there is at most one indecomposable conjugator s starting at y such that $a \preccurlyeq s$. Hence, the number of indecomposable conjugators starting at y is bounded above by the number of atoms of G. This shows in particular the finiteness of the set of arrows of $\mathrm{SCG}(x)$, so the graph is finite. □

4 Applications

4.1 An algorithm to solve the conjugacy problem

One of our main motivations for introducing the concept of cyclic sliding was to simplify the known algorithms to solve the conjugacy decision problem (CDP) and the conjugacy search problem (CSP) in Garside groups of finite type.

We will give in [20] a detailed description of the resulting algorithm, which solves both problems by using cyclic sliding. However, we want at least to give a brief overview of it in this paper. The main idea of the algorithm is very similar to that of the previously known ones [15, 17, 19], only that the use of cyclic sliding makes it more simple.

Basically, given a Garside group G and an element $x \in G$, the algorithm computes the graph $\mathrm{SCG}(x)$. The procedure is the following:

1. Given $x \in G$, apply iterated cyclic sliding until a repeated element $\widetilde{x}$ is obtained. The element $\widetilde{x}$ belongs to $\mathrm{SC}(x)$.

2. For every known element $y \in \mathrm{SC}(x)$, compute all indecomposable conjugators starting at y. Keep track of the obtained conjugators and the resulting conjugates in $\mathrm{SC}(x)$.

   A (very bad) way to do this would be to check, for each simple element s, whether $y^s \in \mathrm{SC}(x)$ (by applying iterated cyclic sliding to y^s), and then to determine among these elements those which are minimal with respect to $\preccurlyeq$. In [20] we shall give a much more efficient procedure to perform this step.

3. Continue with the previous step, until it has been applied to all known elements of $\mathrm{SC}(x)$ and no new elements of $\mathrm{SC}(x)$ are obtained. Since $\mathrm{SCG}(x)$ is finite and connected, this procedures terminates and constructs the entire graph $\mathrm{SCG}(x)$.

The algorithm to solve the CDP and the CSP in a Garside group of finite type then goes as follows. Given $x, y \in G$, compute elements $\widetilde{x} \in \mathrm{SC}(x)$ and $\widetilde{y} \in \mathrm{SC}(y)$ by iterated cyclic sliding as in Step 1 above. Then compute $\mathrm{SCG}(x)$ using the above procedure. If $\widetilde{y}$ is not a vertex of $\mathrm{SCG}(x)$, that is if $\widetilde{y} \notin \mathrm{SC}(x)$, then x and y are not conjugate. Otherwise, using the information about conjugating elements that we obtained during the process, we know a conjugating element from x to $\widetilde{x}$ (the product of preferred prefixes used in iterated cyclic slidings), a conjugating element from $\widetilde{x}$ to $\widetilde{y}$ (a path in the graph $\mathrm{SCG}(x)$), and a conjugating element from $\widetilde{y}$ to y (the inverses of the preferred prefixes that lead from y to $\widetilde{y}$). Concatenating these three elements, one obtains a conjugating element from x to y. This procedure hence solves both problems, CDP and CSP in Garside groups of finite type.

4.2 Rigid elements

The notion of *rigidity* was introduced in [5]. Using the terminology of Section 3.1, we have

Definition 4.1. *An element $x \in G$ is called* **rigid** *if $\mathfrak{p}(x) = 1$.*

Intuitively, $x = \Delta^p x_1 \cdots x_r$ is rigid if the pair $x_r \tau^{-p}(x_1)$ consisting of the last and the first simple factor of x conjugated by Δ^p is left weighted, that is, one has left weightedness of all pairs of consecutive simple factors even when "closing the element x around a circle". The behaviour of such elements is much simpler to understand than in the general case; for example, the only thing necessary to bring a power of x into left normal form is to take care of the powers of Δ. Specifically, $x^k = \Delta^{kp}\tau^{(k-1)p}(x_1 \cdots x_r) \cdots \tau^p(x_1 \ldots x_r) \cdot (x_1 \cdots x_r)$ is in left normal form as written.

We remark that with the above definition, a power of Δ is rigid, while in [5] this was not the case. This is just a convention. However, we think that, using the definition above, it is natural to include powers of Δ in the set of rigid elements.

It was shown in [5, Corollary 3.16] that if an element x is rigid and satisfies $\ell(x) > 1$, then the set of rigid conjugates of x is precisely the *ultra summit set* of x, that is, the set of super summit elements which are in a circuit under cycling. This result, however, does not extend to the case $\ell(x) = 1$.

In this section we show Theorems 1.1 and 1.2. Theorem 1.1 is an analogue of [5, Corollary 3.16] with the ultra summit set replaced by the invariant $\mathrm{SC}(x)$ introduced in Section 3.1. This result, however, does include the case $\ell(x) = 1$ and its proof is much easier than the proof of the result in [5], suggesting that $\mathrm{SC}(x)$ is the more natural invariant to consider.

Definition 4.2. *For $x \in G$ and $i \in \mathbb{N}$ let $\mathfrak{P}_i(x) = \mathfrak{p}(x)\mathfrak{p}(\mathfrak{s}(x)) \cdots \mathfrak{p}(\mathfrak{s}^{i-1}(x))$. (We also define $\mathfrak{P}_0(x) = 1$.) That is, $\mathfrak{P}_i(x)$ is the conjugating element for i-fold cyclic sliding of x.*

Proposition 4.3. *Let x be rigid, let $s \in G$ such that $\inf(x^s) = \inf(x)$ and $\sup(x^s) = \sup(x)$. Then the following hold for all integers $i \geq 0$:*

1. $1 = \mathfrak{P}_0(x^s) \preccurlyeq \mathfrak{P}_1(x^s) \preccurlyeq \cdots \preccurlyeq \mathfrak{P}_i(x^s) \preccurlyeq \mathfrak{P}_{i+1}(x^s) \preccurlyeq \Delta^{\ell(s)}$
2. $s = s^{(0)} \preccurlyeq s^{(1)} \preccurlyeq \cdots \preccurlyeq s^{(i)} \preccurlyeq s^{(i+1)} \preccurlyeq \Delta^{\sup(s)}$

Proof. Denoting $y = x^s$, we have the following commutative diagram in which $\mathfrak{p}(y), \ldots, \mathfrak{p}(\mathfrak{s}^i(y))$ are positive by definition:

$$\begin{tikzcd}
x \arrow[r, "1"] \arrow[d, "s"'] & x \arrow[r, "1"] \arrow[d, "s^{(1)}"'] & \cdots \arrow[r, "1"] & x \arrow[r, "1"] \arrow[d, "s^{(i)}"] & x \arrow[d, "s^{(i+1)}"] \\
y \arrow[r, "\mathfrak{p}(y)"'] & \mathfrak{s}(y) \arrow[r, "\mathfrak{p}(\mathfrak{s}(y))"'] & \cdots \arrow[r, "\mathfrak{p}(\mathfrak{s}^{i-1}(y))"'] & \mathfrak{s}^i(y) \arrow[r, "\mathfrak{p}(\mathfrak{s}^i(y))"'] & \mathfrak{s}^{i+1}(y)
\end{tikzcd}$$

By induction, we in particular have $1 = \mathfrak{P}_0(x^s) \preccurlyeq \mathfrak{P}_1(x^s) \preccurlyeq \cdots \preccurlyeq \mathfrak{P}_i(x^s) \preccurlyeq \mathfrak{P}_{i+1}(x^s)$ and $s = s^{(0)} \preccurlyeq s^{(1)} \preccurlyeq \cdots \preccurlyeq s^{(i)} \preccurlyeq s^{(i+1)}$. Moreover, $s^{(i+1)} \preccurlyeq \Delta^{\sup(s)}$ by Corollary 3.19. As $\Delta^{\inf(s)} \preccurlyeq s$, we thus have $\mathfrak{P}_{i+1}(y) = s^{-1}s^{(i+1)} \preccurlyeq s^{-1}\Delta^{\sup(s)} \preccurlyeq \Delta^{\sup(s)-\inf(s)} = \Delta^{\ell(s)}$ as claimed. □

The following corollary is equivalent to Theorem 1.1.

Corollary 4.4. *If x is rigid, then* $\mathrm{SC}(x)$ *is the set of rigid conjugates of* x.

Proof. Given any rigid conjugate y of x, we have $\mathfrak{s}(y) = y$, whence in particular $y \in \mathrm{SC}(x)$. It remains to be shown that all elements of $\mathrm{SC}(x)$ are rigid. As x itself is rigid, we know that $\mathrm{SC}(x)$ contains at least one rigid element.

Suppose that $y = x^s \in \mathrm{SC}(x)$ is not rigid. As some power of Δ is central, we can assume that s is positive. If $\mathfrak{s}^i(y)$ was rigid for some $i \in \mathbb{N}$, we would have $\mathfrak{s}^j(y) = \mathfrak{s}^i(y) \neq y$ for all $j \geq i$, contradicting the fact that y is in a sliding circuit. Hence $\mathfrak{p}(\mathfrak{s}^i(y)) \neq 1$ for all $i \in \mathbb{N}$ and we obtain an ascending chain $1 \prec \mathfrak{P}_1(y) \prec \mathfrak{P}_2(y) \prec \mathfrak{P}_3(y) \prec \ldots$ where $\mathfrak{P}_i(y) \preccurlyeq \Delta^{\ell(s)}$ for all $i \in \mathbb{N}$ by Proposition 4.3. This is impossible, however, as G is of finite type, that is, there cannot exist a non-rigid element in $\mathrm{SC}(x)$. □

Let us now show Theorem 1.2, with the aid of the following result.

Lemma 4.5. *If x is conjugate to a rigid element, $y \in \mathrm{SSS}(x)$ and $c(y)$ is the minimal positive element such that $y^{c(y)} \in \mathrm{SC}(x)$ as in Corollary 3.34, then $c(y)^{(k)} = c(\mathfrak{s}^k(y))$ for all integers $k \geq 0$.*

Proof. For $k = 0$ there is nothing to show. Since $z = y^{c(y)}$ is rigid by Corollary 4.4, we have $\mathfrak{p}(z) = 1$ and $\mathfrak{s}(z) = z$. Moreover, the diagram

$$\begin{tikzcd}
y \arrow[r, "\mathfrak{p}(y)"] \arrow[d, "c(y)"'] & \mathfrak{s}(y) \arrow[d, "c(y)^{(1)}"] \\
z \arrow[r, "1"'] & z
\end{tikzcd}$$

is commutative, that is, $c(y) = \mathfrak{p}(y)c(y)^{(1)}$ and $c(y)^{(1)}$ is positive by Lemma 3.15. From Corollary 3.34 we then obtain $c(y) \preccurlyeq \mathfrak{p}(y)c(\mathfrak{s}(y))$ and also $c(\mathfrak{s}(y)) \preccurlyeq c(y)^{(1)} = \mathfrak{p}(y)^{-1}c(y) \preccurlyeq c(\mathfrak{s}(y))$ showing the claim for $k = 1$. As $\mathfrak{s}(y) \in \mathrm{SSS}(x)$, the claim then follows by induction. □

Corollary 4.6. *If x is conjugate to a rigid element, $y \in \mathrm{SSS}(x)$ and $c(y)$ is the minimal positive element such that $y^{c(y)} \in \mathrm{SC}(x)$ as in Corollary 3.34, then there exists an integer M such that $c(y) = \mathfrak{P}_i(y)$ for all $i \geq M$.*

Proof. If M is chosen such that $\mathfrak{s}^M(y) \in \mathrm{SC}(x)$, then $\mathfrak{s}^M(y)$ is rigid by Corollary 4.4 and we have $\mathfrak{P}_i(y) = \mathfrak{P}_M(y)$ for all $i \geq M$. Moreover, $c(y)^{(M)} = c(\mathfrak{s}^M(y)) = 1$ by Lemma 4.5 and the claim then follows from $1 = c(y)^{(M)} = \mathfrak{P}_M(y)^{-1}\, c(y)\, \mathfrak{P}_M(y^{c(y)}) = \mathfrak{P}_M(y)^{-1}\, c(y)$. □

According to Corollary 4.6, if a super summit element has rigid conjugates, then the optimal way of obtaining a rigid conjugate through conjugation by positive elements is given by iterated cyclic sliding, so Theorem 1.2 is shown. This indicates that cyclic sliding is a very natural operation.

We remark that if x is not conjugate to a rigid element, then iterated cyclic sliding does not necessarily yield the shortest conjugating element from x to an element in $\mathrm{SC}(x)$. An example is the 4-braid $x = \sigma_3\sigma_2\sigma_1 \in B_4$. One easily checks that $\mathfrak{p}(x) = \sigma_3\sigma_2$ and $\mathfrak{s}(x) = \sigma_1\sigma_3\sigma_2 = \mathfrak{p}(\mathfrak{s}(x))$, whence $\mathfrak{s}^i(x) = \mathfrak{s}(x) \neq x$ for $i \geq 1$. Moreover, $x^{\sigma_3} = \sigma_2\sigma_1\sigma_3 = \mathfrak{p}(x^{\sigma_3})$, whence $\mathfrak{s}^i(x^{\sigma_3}) = x^{\sigma_3}$ for $i \geq 0$. (Note that, while $\mathfrak{s}(x)$ and x^{σ_3} are fixed under cyclic sliding, these elements are not rigid!) In particular, $x \notin \mathrm{SC}(x)$ and $x^{\sigma_3} \in \mathrm{SC}(x)$, that is, $c(x) = \sigma_3 \prec \sigma_3\sigma_2 = \mathfrak{p}(x)$. Moreover, the chain $1 \prec \mathfrak{P}_1(x) \prec \mathfrak{P}_2(x) \prec \ldots$ is not bounded in this case and indeed is an infinite strictly ascending chain.

4.3 Reducible braids

As we mentioned in the introduction, cyclic sliding and the sets of sliding circuits satisfy in a natural way the good properties that were known for cycling, decycling and ultra summit sets. Some of these properties, in the particular case of braid groups, concern reducible braids. This is one of the important aspects of the project to solve the CDP/CSP in braid groups in polynomial time which is described in [5].

Considering braids in B_n as automorphisms of the n-times punctured disc D_n, up to isotopy fixing the boundary, a braid is called reducible if it preserves setwise a family of disjoint closed simple curves in D_n, each one enclosing more than 1 and less than n punctures. These curves are known as reducing curves of the corresponding braid. There is a special family of reducing curves associated to each reducible braid, called its *canonical reduction system* [10]. If one is able to detect efficiently the canonical reduction system of a braid, the CDP/CSP can be split into simpler problems, as explained in [5]. In any case, an efficient way of computing the reducing curves of a braid would lead to an efficient geometric classification of the braid into reducible, periodic or pseudo-Anosov.

There are two well known algorithms to determine whether a braid is reducible by computing reducing curves. The first one is due to Bestvina and Handel [4] and can be applied not only to braids but also to automorphisms of any compact surface. But the complexity of this algorithm does not seem to be polynomial, and to our knowledge it has not been studied, even in the particular case of braid groups. The second algorithm was given by Benardete, Gutiérrez and Nitecki [2, 3], and uses the Garside structure of braid groups.

A reducing curve is said to be *standard* if it is isotopic to a geometric circle, or equivalently, if the punctures that it encloses are consecutive (we assume the punctures to be placed on a line). The main result in [3] states that if $x \in B_n$ admits a standard reducing curve $\mathcal{C}$, and $\Delta^p x_1 \cdots x_r$ is the left normal form of x, then the image of $\mathcal{C}$ under $\Delta^p x_1 \cdots x_i$ is also standard, for $i = 0, \ldots, r$. This implies, in particular, that if x admits a standard reducing curve, then $\mathbf{c}(x)$ and $\mathbf{d}(x)$ admit standard reducing curves. Since it is clear that every reducible braid x has a conjugate which admits a standard reducing curve, iterated application of cyclings and decyclings to that conjugate yields that in $\mathrm{SSS}(x)$ there is an element which admits a standard reducing curve. Since $\mathrm{SSS}(x)$ is a finite set, and it is an easy (and finite) procedure to check whether a braid admits a standard reducing curve, this produces an algorithm to find reducing curves for a braid, at the cost of computing the super summit set.

With the introduction of ultra summit sets in [19], it became clear that one does not need to compute the whole super summit set. Starting with an element in $\mathrm{SSS}(x)$ that admits a standard reducing curve, iterated cycling until the first repetition is encountered produces an element in $\mathrm{USS}(x)$, which also admits a standard reducing curve. Hence, an element x is reducible if and only if some element in $\mathrm{USS}(x)$ admits a standard reducing curve. This was a major advance, since ultra summit sets are in general much smaller than super summit sets.

Now recall from Lemma 3.8 that a cyclic sliding can always be expressed as the composition of τ, cycling and decycling. Clearly, τ sends standard reducing curves to standard reducing curves, and by [3], this is also true for cycling and decycling. Therefore one has:

Lemma 4.7. *If a braid $x \in B_n$ admits some standard reducing curve, so does $\mathfrak{s}(x)$.*

Corollary 4.8. *A braid $x \in B_n$ is reducible if and only if there is some element in* $\mathrm{SC}(x)$ *which admits a standard reducing curve.*

5 Examples

In this final section we shall provide some examples, some of them of a theoretical nature and others obtained by computer calculations. They will give some evidence to our assertion that the sets of sliding circuits are substantially better invariants than ultra summit sets, at least for elements of canonical length one. But on the other hand, we will also see that even in the braid groups B_n there are families of elements whose sets of sliding circuits grow exponentially in n. This shows that, although cyclic sliding is a natural choice, some more work remains to be done when trying to find a polynomial algorithm for the conjugacy problem in braid groups.

Let us start with the bad news.

5.1 Exponential sets of sliding circuits

In [7], the authors and Joan S. Birman showed that the number of elements in the ultra summit sets of some periodic braids in B_n is exponential in n. More precisely, $|\mathrm{USS}(\delta)| = 2^{n-2}$, where $\delta = \sigma_{n-1} \cdots \sigma_1 \in B_n$. To overcome this difficulty, in [7] we also gave a polynomial algorithm to solve the conjugacy search problem for all periodic braids (which of course does not involve computing the whole ultra summit set).

Since $\mathrm{SC}(x)$ is contained in $\mathrm{USS}(x)$ for every $x \in B_n$, and it is in general smaller for elements of canonical length one (like δ above), one may wonder whether $\mathrm{SC}(\delta)$ has polynomial size, allowing to use the general algorithm given in this paper (and in [20]) instead of the particular one given in [7], which only works for periodic braids. Unfortunately the answer is negative: There are only two elements which are in $\mathrm{USS}(\delta)$ but not in $\mathrm{SC}(\delta)$, as it is shown in the following result.

Proposition 5.1. *Let $\delta = \sigma_{n-1} \cdots \sigma_1 \in B_n$. One has $|\mathrm{SC}(\delta)| = 2^{n-2} - 2$.*

Proof. In [7, Proposition 10] one can find a characterisation of the elements of $\mathrm{USS}(\delta)$: They are those simple braids s whose associated permutation is a single cycle of length n of the form $\pi_s = (1\ u_1\ u_2\ \cdots\ u_r\ n\ d_t\ d_{t-1}\ \cdots\ d_1)$, where $u_1 < u_2 < \cdots < u_r$ and $d_t > d_{t-1} > \cdots > d_1$. It follows, as noticed in [7], that $|\mathrm{USS}(\delta)| = 2^{n-2}$. There are two special elements in this set: When $r = 0$ one has $\pi_s = (1\ n\ n-1\ \cdots\ 2) = (n\ n-1\ \cdots\ 1)$, whence $s = \sigma_1 \cdots \sigma_{n-1}$, and when $t = 0$ one has $\pi_s = (1\ 2\ \cdots\ n)$, whence $s = \sigma_{n-1} \cdots \sigma_1 = \delta$. We will show shortly that these two elements do not belong to $\mathrm{SC}(\delta)$, but first we will see that all other elements in $\mathrm{USS}(\delta)$ do. Hence, we will assume for a moment that s is a simple element whose permutation has the above form, with $r, t > 0$. We claim that, in this situation, s^2 is simple.

In order to show the above claim, we just need to prove that for every distinct $i, j \in \{1, \ldots, n\}$, the strands i and j cross at most once in s^2. Notice that, as s is simple, every two strands cross at most once in s, so they can cross at most twice in s^2. Recall that i and j (with $i < j$) cross in s if and only if $\pi_s(i) > \pi_s(j)$. Hence, the claim is false if and only if there are $i, j \in \{1, \ldots, n\}$ such that $i < j,\ \ \pi_s(i) > \pi_s(j)$ and $\pi_s^2(i) < \pi_s^2(j)$.

Let $U = \{1, u_1, \ldots, u_r\}$ be the set of punctures that 'move to the right' in s, and $D = \{d_1, \ldots, d_t, n\}$ the set of punctures that 'move to the left'. Notice that if two punctures i and j with $i < j$ cross in s, then $i \in U$ and $j \in D$. Moreover, after the crossing, $\pi_s(j) < \pi_s(i)$. Hence, if $\pi_s(j)$ and $\pi_s(i)$ cross again in s, one must have $\pi_s(j) \in U$ and $\pi_s(i) \in D$. But the only puncture in U whose image under π_s belongs to D is u_r, and the only puncture in D whose image under π_s belongs to U is d_1. Therefore, if i and j with $i < j$ cross twice in s^2, one must have $i = u_r$ and $j = d_1$. This would imply that $u_r < d_1$, and this can only happen if $(1, \ldots, n) = (1, u_1, \ldots, u_r, d_1, \ldots, d_t, n)$. But in this case, since we are assuming that $r, t > 0$, it follows that the strands $\pi_s(u_r) = n$ and $\pi_s(d_1) = 1$ do not cross in s, which is a contradiction. Hence, the claim is true.

We have shown that if $r, t > 0$, then s^2 is a simple braid. This implies that $\mathfrak{p}(s) = s$ and then $\mathfrak{s}(s) = s^s = s$, hence $s \in \mathrm{SC}(\delta)$. This shows that all elements of $\mathrm{USS}(\delta)$, except possibly $\sigma_1 \cdots \sigma_{n-1}$ and $\sigma_{n-1} \cdots \sigma_1$, belong to $\mathrm{SC}(\delta)$. That is, $2^{n-2} - 2 \leq |\mathrm{SC}(\delta)| \leq 2^{n-2}$.

Finally, the left normal form of $(\sigma_1 \cdots \sigma_{n-1})^2$ is $(\sigma_1 \cdots \sigma_{n-1}\sigma_1 \cdots \sigma_{n-2})\sigma_{n-1}$, whence we have $\mathfrak{p}(\sigma_1 \cdots \sigma_{n-1}) = \sigma_1 \cdots \sigma_{n-2}$ and $\mathfrak{s}(\sigma_1 \cdots \sigma_{n-1}) = (\sigma_1 \cdots \sigma_{n-1})^{\sigma_1 \cdots \sigma_{n-2}} = \sigma_{n-1}\sigma_1 \cdots \sigma_{n-2}$. The associated permutation of the latter braid is $(1\ n-1\ n\ n-2\ n-3\ \cdots\ 2)$. Therefore, by the previous paragraph, $\sigma_1 \cdots \sigma_{n-1} \neq \mathfrak{s}(\sigma_1 \cdots \sigma_{n-1}) = \mathfrak{s}^i(\sigma_1 \cdots \sigma_{n-1})$ for all $i \geq 1$, implying that $\sigma_1 \cdots \sigma_{n-1} \notin \mathrm{SC}(\delta)$. Analogously, the left normal form of δ^2 is $(\sigma_{n-1} \cdots \sigma_1 \sigma_{n-1} \cdots \sigma_2)\sigma_1$, so $\mathfrak{p}(\delta) = \sigma_{n-1} \cdots \sigma_2$ and $\mathfrak{s}(\delta) = \sigma_1 \sigma_{n-1} \cdots \sigma_2$, whose associated permutation is $(1\ 3\ 4\ \cdots\ n\ 2)$. Therefore, $\delta \neq \mathfrak{s}(\delta) = \mathfrak{s}^i(\delta)$ for all $i \geq 1$, which implies that $\delta \notin \mathrm{SC}(\delta)$. We have thus shown that $|\mathrm{SC}(\delta)| = 2^{n-2} - 2$ as claimed. □

5.2 Comparison between sliding circuits and ultra summit sets

We now present the results of computer experiments, in which we compare the sizes of ultra summit sets and sets of sliding circuits in braid groups. First we notice that, after the computations shown in [19], random braids of large canonical length have rigid conjugates with an overwhelming probability (100% of thousands of cases). If x is conjugate to a rigid element, we showed in Theorem 1.1 that $\mathrm{SC}(x)$ is the set of rigid conjugates of x. If furthermore $\ell(x) > 1$, it is shown in [5] that $\mathrm{USS}(x)$ is also the set of rigid conjugates of x. This implies that, if one computed random examples of braids of large canonical length, one would in virtually all cases have $\mathrm{USS}(x) = \mathrm{SC}(x)$ with the size of this set equal to $2\ell(x)$, as noticed in [19]. Finding braids with large ultra summit sets is a difficult problem, unless one uses families of examples like the one shown in the previous subsection. Hence, random computations with braids of large canonical length would not lead to a meaningful comparison between sets of sliding circuits and ultra summit sets.

The other extreme situation, the case of elements of canonical length 1, is different. If $\ell(x) = 1$, then one has $\mathrm{SSS}(x) = \mathrm{USS}(x)$, and even if x has a rigid conjugate, it is not necessarily true that $\mathrm{USS}(x) = \mathrm{SC}(x)$. One can then expect to see substantial differences between the sizes of $\mathrm{SC}(x)$ and $\mathrm{USS}(x) = \mathrm{SSS}(x)$ in most cases. Therefore, as the case of canonical length 1 is the only interesting of the extreme cases as far as computational experiments are concerned, we have made computations with braids of canonical length 1.

We remark that if one picks random braids, the larger conjugacy classes are more likely to appear, which can alter the conclusions. Therefore, we decided to perform exhaustive tests rather than use random braids: For $n = 4, \ldots, 12$, we computed the super (=ultra) summit sets and the sets of sliding circuits for all conjugacy classes with summit infimum 0 and summit canonical length 1 in B_n, with the usual Garside structure. The results are shown in Figure 2.

	# of conj.	Maximum			Mean among conj. classes			Mean among elements		
n	classes	$\|\mathrm{SSS}\|$	$\|\mathrm{SC}\|$	$\frac{\|\mathrm{SSS}\|}{\|\mathrm{SC}\|}$	$\|\mathrm{SSS}\|$	$\|\mathrm{SC}\|$	$\frac{\|\mathrm{SSS}\|}{\|\mathrm{SC}\|}$	$\|\mathrm{SSS}\|$	$\|\mathrm{SC}\|$	$\frac{\|\mathrm{SSS}\|}{\|\mathrm{SC}\|}$
4	9	4	4	2	2.44444	2.22222	1.11111	3.09091	2.72727	1.18182
5	26	12	8	6	4.53846	3.30769	1.42308	6.57627	4.0678	1.87571
6	89	38	22	15	8.06742	4.40449	2.14131	16.2646	6.38162	3.78721
7	305	142	58	60	16.518	5.91475	3.52468	48.7674	10.3355	8.52684
8	1278	650	120	208	31.5477	6.83255	6.25794	154.13	15.3566	22.2361
9	6096	3228	528	882	59.5272	7.41503	11.702	548.184	23.9919	80.0996
10	35631	18226	1664	5900	101.844	7.12862	20.7484	2046.23	40.2408	299.891
11	244127	97762	4564	33432	163.508	6.3462	34.3723	7863.68	64.9602	1061.36
12	1940201	651528	28026	200172	246.882	5.46008	54.0114	31252.1	109.12	4016.81

Figure 2: Sizes of $\mathrm{SSS}(x)$ and $\mathrm{SC}(x)$ for conjugacy classes with summit infimum 0 and summit canonical length 1 in B_n.

We can see how the maximal and mean values of $\mathrm{SSS}(x)$ and $\mathrm{SC}(x)$ change as n grows. For example, choosing one of the 1940201 conjugacy classes with summit infimum 0 and summit canonical length 1 in B_{12} at random (with uniform probability on the set of conjugacy classes), the expected value for the ratio of the size of its super summit set and the size of its set of sliding circuits is about 54. On the other hand, choosing one of the $12! - 2$ elements with infimum 0 and canonical length 1 in B_{12} at random (with uniform probability on the set of elements), the expected value for the ratio of the size of its super summit set and the size of its set of sliding circuits is about 4016. This difference between class mean and element mean tells us that the difference between the size of the super summit set and the size of the set of sliding circuits tends to be more significant for larger super summit sets than for smaller ones.

There are other elements with canonical length 1 in B_n with the usual Garside structure, besides those with infimum 0, namely those of the form $\Delta^m s$ where s has infimum 0 and canonical length 1. Since Δ^2 is central, one has $\mathrm{SSS}(\Delta^{2k+p}s) = \Delta^{2k}\,\mathrm{SSS}(\Delta^p s)$ and $\mathrm{SC}(\Delta^{2k+p}s) = \Delta^{2k}\,\mathrm{SC}(\Delta^p s)$ for every $k \in \mathbb{Z}$. In particular, it is sufficient to consider the cases $p = 0$ (see Figure 2) and $p = 1$. For the case $p = 1$, note that $(\Delta s)^c = \Delta t$ is equivalent to $((\Delta s)^{-1})^c = (\Delta t)^{-1}$, that is, $(s^{-1}\Delta^{-1})^c = t^{-1}\Delta^{-1}$, which in turn is equivalent to $(s^{-1}\Delta)^c = t^{-1}\Delta$, that is, $\partial(s)^c = \partial(t)$, since Δ^2 is central. Moreover, $\mathfrak{p}(\partial(s)) = \mathfrak{p}(s^{-1}\Delta) = \mathfrak{p}(\Delta^{-1}s) = \mathfrak{p}(\Delta s)$. Thus, the bijective map

$$\mu : \{x \in B_n : \inf(x) = 1,\, \ell(x) = 1\} \to \{x \in B_n : \inf(x) = 0,\, \ell(x) = 1\}$$

defined by $\mu(\Delta s) = \partial(s)$ respects conjugacy and induces isomorphisms of sliding circuit graphs. In particular, $|\mathrm{SSS}(\Delta s)| = |\mathrm{SSS}(\partial(s))|$ and $|\mathrm{SC}(\Delta s)| = |\mathrm{SC}(\partial(s))|$. Hence, the classes with odd summit infimum and summit canonical length 1 give the same results as in Figure 2.

We did analogous computations for the n-strand braid groups using the Birman-Ko-Lee (BKL) Garside structure [8]; we denote these Garside groups by BKL_n. The Garside element of BKL_n is $\delta = \sigma_{n-1}\sigma_{n-2}\cdots\sigma_1$, and $\delta^n = \Delta^2$ is central. Similarly to above, we have for every s with infimum 0 and canonical length 1 (with respect to the BKL structure) and every $k \in \mathbb{Z}$ that $\mathrm{SSS}(\delta^{kn+i}s) = \delta^{kn}\,\mathrm{SSS}(\delta^i s)$ and $\mathrm{SC}(\delta^{kn+i}s) = \delta^{kn}\,\mathrm{SC}(\delta^i s)$, so we just need to study the conjugacy classes with summit canonical length 1 and summit infimum i for $i = 0, \ldots, n-1$. Again similarly to above, notice that $(\delta^i s)^c = \delta^i t$ is equivalent to $(\delta^{n-i-1}\tau^{n-i-1}(\partial(s)))^c = \delta^{n-i-1}\tau^{n-i-1}(\partial(t))$ and $\mathfrak{p}(\delta^i s) = \mathfrak{p}(\delta^{n-i-1}\tau^{n-i-1}(\partial(s)))$, where τ denotes conjugation by the Garside element δ. Hence, the bijective map

$$\nu : \{x \in BKL_n : \inf(x) = i,\, \ell(x) = 1\} \to \{x \in BKL_n : \inf(x) = n-i-1,\, \ell(x) = 1\}$$

defined by $\nu(\delta^i s) = \delta^{n-i-1}\tau^{n-i-1}(\partial(s))$ respects conjugacy and induces isomorphisms of sliding circuit graphs. In particular, the classes with summit infimum i and summit canonical length 1 will give the same results as the classes with summit infimum $n-i-1$ and summit canonical length 1, whence we just need to study the cases $0 \le i < n/2$. The results are given in Figure 3.

We point out that for every simple element s with respect to the BKL structure, $\mathrm{SSS}(s) = \mathrm{SC}(s)$. This is due to the fact that every simple element (with respect to the BKL structure) is rigid. Also, we can see from Figure 3 that if n is odd, $\mathrm{SSS}(\delta^{\frac{n-1}{2}}s) = \mathrm{SC}(\delta^{\frac{n-1}{2}}s)$ for every simple element s. (However, the elements $\delta^{\frac{n-1}{2}}s$, with s simple, are in general not rigid!) For other values of the summit infimum i, however, we see that the difference between the sizes of $\mathrm{SSS}(\delta^i s)$ and $\mathrm{SC}(\delta^i s)$ increases as n grows.

We finish by giving a couple of particular examples. In Figure 2 we see that in B_{12}, the maximal ratio $|\mathrm{SSS}(x)|/|\mathrm{SC}(x)|$ is 200172. This value is obtained by the simple element whose induced permutation is (1 3 10 12 2 5 4 7 8 9 11). In standard generators:

$$x = \sigma_{10}\sigma_9\sigma_8\sigma_7\sigma_6\sigma_5\sigma_4\sigma_3\sigma_2\sigma_1\ \sigma_{11}\sigma_{10}\sigma_9\sigma_8\sigma_7\sigma_6\sigma_5\sigma_4\sigma_3\sigma_2\ \sigma_6\sigma_5\sigma_4\ \sigma_7\sigma_6\ \sigma_7\ \sigma_8\ \sigma_9.$$

In this case one has $|\mathrm{SSS}(x)| = 400344$, while $|\mathrm{SC}(x)| = 2$.

The last example was not obtained from our computations, but from a theoretical result. In [21], Pedro González Manchón gave an example of two simple braids in B_{12} which are not conjugate,

		# of conj.	Maximum			Mean among conj. classes			Mean among elements		
n	i	classes	|SSS|	|SC|	$\frac{\|SSS\|}{\|SC\|}$	|SSS|	|SC|	$\frac{\|SSS\|}{\|SC\|}$	|SSS|	|SC|	$\frac{\|SSS\|}{\|SC\|}$
6	0	9	30	30	1	14.4444	14.4444	1	21.2	21.2	1
6	1	18	24	18	4	7.22222	5.22222	1.44444	11.5538	6.84615	1.92308
6	2	16	24	18	4	8.125	6.25	1.3125	13.3538	8.36923	1.83077
7	0	13	105	105	1	32.8462	32.8462	1	54.5082	54.5082	1
7	1	31	63	28	9	13.7742	9.03226	1.64516	22.8361	10.4426	2.70492
7	2	29	42	28	5	14.7241	10.3793	1.47701	23.2951	13.4262	2.02186
7	3	26	42	42	1	16.4231	16.4231	1	26.9672	26.9672	1
8	0	20	280	280	1	71.4	71.4	1	141.518	141.518	1
8	1	72	128	40	16	19.8333	10.3333	1.98843	41.7311	13.3838	3.85994
8	2	73	120	80	9	19.5616	14.137	1.56176	43.3445	27.5686	2.12899
8	3	55	136	72	17	25.9636	15.6364	1.85455	56.4314	22.1008	3.91877
9	0	28	756	756	1	173.571	173.571	1	384.748	384.748	1
9	1	146	225	72	25	33.2877	15.1027	2.50742	82.4222	20.9389	5.56728
9	2	159	297	90	25	30.566	13.2453	2.64937	83.2481	17.4148	6.80417
9	3	128	369	90	13.6667	37.9688	17.7891	2.18232	101.181	27.5981	4.35071
9	4	102	432	432	1	47.6471	47.6471	1	130.878	130.878	1
10	0	40	2520	2520	1	419.85	419.85	1	1078.74	1078.74	1
10	1	342	660	120	40	49.1053	19.1053	2.76831	165.045	32.39	7.16021
10	2	405	610	120	61	41.4667	14.479	2.9339	149.086	23.5297	8.56712
10	3	344	650	270	53	48.8198	23.3547	2.65378	182.587	58.5601	6.92947
10	4	219	760	240	76	76.6849	34.4475	2.821	264.128	65.4733	8.90691
11	0	54	6930	6930	1	1088.59	1088.59	1	3100.34	3100.34	1
11	1	775	1870	209	75	75.8503	26.4142	3.10328	338.234	53.3512	9.14594
11	2	1019	1782	308	162	57.6879	18.7399	3.52813	275.543	31.7382	15.8496
11	3	912	1958	704	60	64.4561	30.5154	2.82067	329.329	125.985	6.52906
11	4	619	1793	352	72	94.9661	27.1179	4.0453	411.088	50.8565	12.5776
11	5	491	2970	2970	1	119.723	119.723	1	617.622	617.622	1

Figure 3: Sizes of $\mathrm{SSS}(x)$ and $\mathrm{SC}(x)$ for conjugacy classes with summit infimum i and summit canonical length 1 in BKL_n.

but whose associated permutations are centrally conjugate (a notion related to the coefficients of those permutations in the expressions of elements in the centre of the Hecke algebra). These two braids arose from work by T. Hall and A. de Carvalho (see [21]). One of them is:

$$x = \sigma_7\sigma_8\sigma_7\sigma_6\sigma_5\sigma_4\sigma_9\sigma_8\sigma_7\sigma_6\sigma_5\sigma_4\sigma_3\sigma_2\sigma_{10}\sigma_9\sigma_8\sigma_7\sigma_6\sigma_5\sigma_4\sigma_3\sigma_2\sigma_1\sigma_{11}\sigma_{10}\sigma_9\sigma_8\sigma_7\sigma_6\sigma_5\sigma_4\sigma_3\sigma_2\sigma_1.$$

The two mentioned braids were shown not to be conjugate using the algorithm in [19], that is, computing the ultra summit set of x. But $|\mathrm{USS}(x)| = |\mathrm{SSS}(x)| = 126498$. Using our new method one finds that $|\mathrm{SC}(x)| = 6$, so it is almost immediate (and could even be done by hand) to check whether x is conjugate to another braid.

These are just two examples of the possible difference between ultra summit sets and sets of sliding circuits, although we would like to finish by recalling that our motivation for introducing this new tool is mainly theoretical, since we believe that it is a more natural notion for studying conjugacy in Garside groups.

References

[1] S. I. Adyan, *Fragments of the word Δ in the braid group*, (Russian) Mat. Zametki **36** (1984), 25-34.

[2] D. Benardete, M. Gutierrez and Z. Nitecki, *A combinatorial approach to reducibility of mapping classes*, Contemporary Math. **150** (1993), 1-31.

[3] D. Benardete, M. Gutierrez and Z. Nitecki, *Braids and the Nielsen-Thurston classification*, J. Knot Theory and its Ramif. **4** (1995), 549-618.

[4] M. Bestvina and M. Handel, *Train-tracks for surface homeomorphisms*, Topology **34** (1995), 109-140.

[5] J. S. Birman, V. Gebhardt and J. González-Meneses, *Conjugacy in Garside groups I: cycling, powers and rigidity*, Groups Geom. Dyn. **1** (2007), 221-279.

[6] J. S. Birman, V. Gebhardt and J. González-Meneses, *Conjugacy in Garside groups II: Structure of the ultra summit set*, Groups Geom. Dyn. **2** (2008), 16-31.

[7] J. S. Birman, V. Gebhardt and J. González-Meneses, *Conjugacy in Garside groups III: Periodic braids*, J. Alg. **316** (2007), 746-776.

[8] J. S. Birman, K. Y. Ko and S. J. Lee, *A new approach to the word and conjugacy problems in the braid groups*, Adv. Math. **139** (1998), 322-353.

[9] J. S. Birman, K. Y. Ko and S. J. Lee, *The infimum, supremum and geodesic length of a braid conjugacy class*, Adv. Math. **164** (2001), 41-56.

[10] J. S. Birman, A. Lubotzky and J. McCarthy, *Abelian and solvable subgroups of the mapping class groups*, Duke Math. **50** (1983), 1107-1120.

[11] R. Charney, *Artin groups of finite type are biautomatic*, Math. Ann. **292** (1992), 671-683.

[12] P. Dehornoy and L. Paris, *Gaussian groups and Garside groups, two generalizations of Artin groups*, Proc. London Math. Soc. **79** (1999), 569-604.

[13] P. Dehornoy, *Groupes de Garside*, Ann. Scient. Ec. Norm. Sup. **35** (2002), 267-306.

[14] P. Deligne, *Les immeubles des groupes de tresses généralisés*, Invent. Math. **17** (1972), 273-302.

[15] E. ElRifai and H. Morton, *Algorithms for positive braids*, Quart. J. Math. Oxford Ser. (2), **45** (1994), 479-497.

[16] D. B. A. Epstein, J. W. Cannon, D. F. Holt, S. .V. F. Levy, M. S. Patterson and W. P. Thurston, *Word Processing in Groups*, Jones and Bartlett Publishers, Boston 1992.

[17] N. Franco and J. González-Meneses, *Conjugacy problem for braid groups and Garside groups*, J. Alg. **266** (2003), 112-132.

[18] F. Garside, *The braid group and other groups*, Quart. J. Math. Oxford Ser. (2), **20** (1969), 235-254.

[19] V. Gebhardt, *A new approach to the conjugacy problem in Garside groups*, J. Alg. **292** (2005), 282-302.

[20] V. Gebhardt and J. González-Meneses, *Solving the conjugacy problem in Garside groups by cyclic sliding*, arXiv:0809.0948v1.

[21] P. González Manchón, *There exist conjugate simple braids whose associated permutations are not strongly conjugate*, Math. Proc. Cambridge Phil. Soc. **143** (2007), 663-667.

[22] E.-K. Lee and S. J. Lee, *Abelian subgroups of Garside groups*, Comm. Alg. **36** (2008), 1121-1139.

[23] E.-K. Lee and S. J. Lee, *A Garside-theoretic approach to the reducibility problem in braid groups*, J. Alg. **320** (2008), 783-820.

[24] E.-K. Lee and S. J. Lee, *Some power of an element in a Garside group is conjugate to a periodically geodesic element*, Bull. London Math. Soc. **40** (2008), 593-603.

[25] S. J. Lee, *Algorithmic solutions to decision problems in the braid groups*, PhD thesis, Korea Advanced Institute of Science and Technology, 2000.

[26] H. Zheng, *General cycling operations in Garside groups*, arXiv:math/0605741v2 [math.GT].

Volker Gebhardt:
School of Computing and Mathematics
University of Western Sydney
Locked Bag 1797
Penrith South DC NSW 1797, Australia
E-mail: v.gebhardt@uws.edu.au

Juan González-Meneses:
Dept. Álgebra. Facultad de Matemáticas
Universidad de Sevilla
Apdo. 1160
41080 Sevilla (SPAIN)
E-mail: meneses@us.es
URL: www.personal.us.es/meneses